SQL – DİĞER Deyimler ve Komutlar

- İÇİ İÇE SELECT KULLANIMI (NESTED SELECT)
- JOIN KULLANIMI
- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

by yselim

SQL – İç İçe Select (Nested Select)

- **İç içe Select sorguları**, bir başka deyişle **Alt Sorgu** kavramı **SQL**'de en zorlanılan kısımdır.
- Bir sorgudan elde ettiğiniz sonucu, diğer bir sorgu için kullanmanız gerektiğinde, iç içe sorgu kullanmanız gerekir.
- **Alt sorgu** kavramı, genellikle birden fazla tablo söz konusu ise ortaya çıkar.

SQL – İç İçe Select (Nested Select)

Tablo birleştirme işlemini 2 farklı yöntem ile yapabilirsiniz ;

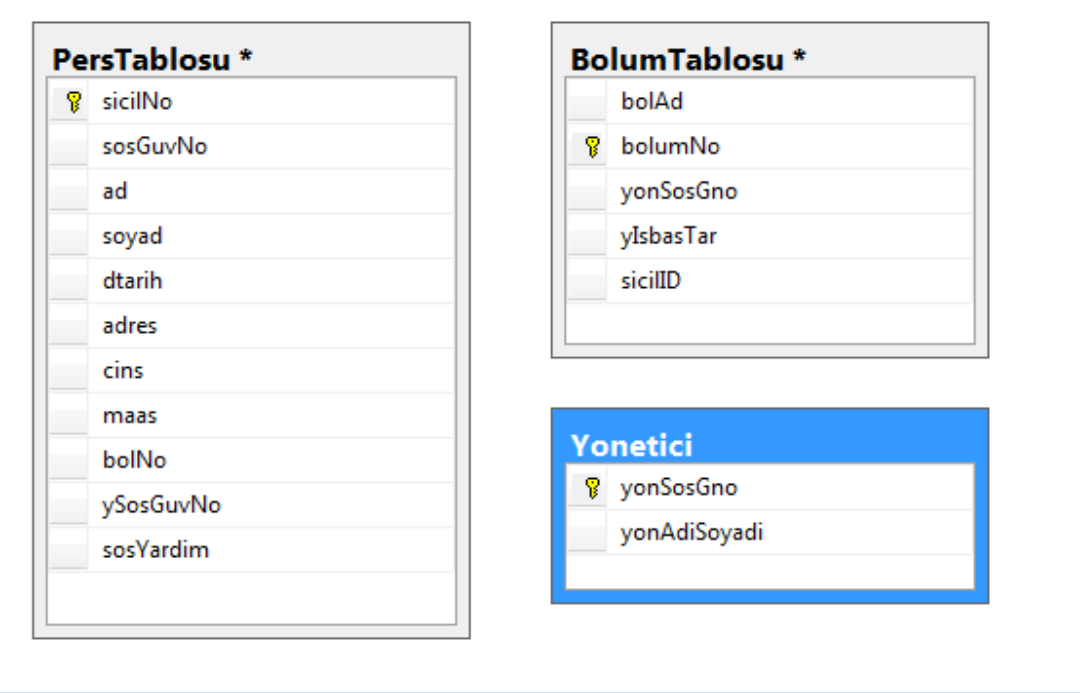
- **WHERE ile sorgu koşulu belirleyerek Tablo Birleştirme**
- **JOIN komutlarını kullanarak Tablo Birleştirme**

SQL – İç İçe Select (Nested Select)

- Eşit değil işareti <>
- Sorular: (ÖĞRENCi-DERSLER tablosu için sorular)
- En yüksek not nedir bulunuz?
- En yüksek notu alan öğrencinin adı soyadı,notu?
- Bölümlere göre en yüksek not değerleri nelerdir?
- Bölümlere göre en yüksek not değerleri alan öğrenciler ve aldıkları dersler?
- Her bölümde en yüksek not alan öğrenciler ve bölümleri?
- Enerji bölümü dışındaki bölümlerdeki en yüksek not alanlar kimler?

SQL – İç İçe Select (Nested Select)

Aşağıda şeması verilmiş tablolar için soruları cevaplayarak iç içe select kullanımı pekiştirelim.



by yselim

SQL – İç İçe Select (Nested Select)

Aşağıda şeması verilmiş tablolar için soruları cevaplayarak iç içe select kullanımı pekiştirelim.

Bolum tablosu

	bolAd	bolumNo	yonSosGno	ylsbasTar	sicilID
1	Satis	1	0371	1989-01-07	5
2	Muhasebe	2	2771	1991-02-08	33
3	Uretim	3	5772	1995-05-05	55
4	elektrik	4	2772	2005-05-06	175

Yonetici tablosu

	yonSosGno	yonAdiSoyadi
1	0371	Ali Kulak
2	2771	Caner Ardıç
3	2772	Sanem Durdu
4	5772	Necati Yavaş

Personel tablosu

	sicilNo	sosGuvNo	ad	soyad	dtarih	adres	cins	maas	bolNo	ySosGuvNo	sosYardim
1	5	6764	Ali	Can	1960-05-01	Karabas m. izmit	E	7600	1	0371	E
2	33	2764	Serhat	Can	1960-05-01	Karabas m. izmit	E	6666	3	5772	E
3	55	6764	Naim	Can	1960-05-01	Karabas m. izmit	E	6600	2	2771	E
4	112	2764	Berk	Can	1960-05-01	Karabas m. izmit	E	8800	4	2772	H
5	175	9674	Aliye	Canan	1970-05-01	sakarya	K	6000	1	0371	E
6	217	1762	Akin	Oncel	1965-07-11	izmit	E	7700	2	2771	H
7	223	0000	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0371	NULL

by yselim

	sicilNo	sosGuvNo	ad	soyad	dtarih	adres	cins	maas	bolNo	ySosGuvNo	sosYardim
1	5	6764	Ali	Can	1960-05-01	Karabas m. izmit	E	7600	1	0371	E
2	33	2764	Serhat	Can	1960-05-01	Karabas m. izmit	E	6666	3	5772	E
3	55	6764	Naim	Can	1960-05-01	Karabas m. izmit	E	6600	2	2771	E
4	112	2764	Berk	Can	1960-05-01	Karabas m. izmit	E	8800	4	2772	H
5	175	9674	Aliye	Canan	1970-05-01	sakarya	K	6000	1	0371	E
6	217	1762	Akin	Oncel	1965-07-11	izmit	E	7700	2	2771	H
7	223	0000	NULL	NULL	NULL	NULL	NULL	NULL	NULL	0371	NULL

	bolAd	bolumNo	yonSosGno	ylsbasTar	sicilID
1	Satis	1	0371	1989-01-07	5
2	Muhasebe	2	2771	1991-02-08	33
3	Uretim	3	5772	1995-05-05	55
4	elektrik	4	2772	2005-05-06	175

	yonSosGno	yonAdiSoyadi
1	0371	Ali Kulak
2	2771	Caner Ardıç
3	2772	Sanem Durdu
4	5772	Necati Yavaş

- Sorular:
- En yüksek maaş değerini bulunuz?
- En yüksek maaşı alan personeli bulunuz?
- Bölümlere göre en yüksek maaş değerleri?
- Her bölümde en yüksek maaş alan çalışanların isimleri?
- 1. bölüm dışındaki bölümlerde en yüksek maaş alanlar kimler?
- En yüksek maaşı alan personelin adı maaşı ve yöneticisinin ismi?
- En düşük maaşı alan personelin yöneticisinin bölümü?

by yselim

SQL – İç İçe Select (Nested Select)

Cevaplar:

- **En yüksek maaş değerini bulunuz?**

```
select max(maas) from PersTablosu
```

- **En yüksek maaşı alan personeli bulunuz?**

```
select ad,soyad,maas from PersTablosu where maas=(select max(maas) from PersTablosu)
```

- **Bölümlere göre en yüksek maaşlar?**

```
select max(maas),bolNo from PersTablosu group by bolNo
```

- **Her bölümde en yüksek maaş alan personeller kimler?**

```
select ad,maas,bolNo from PersTablosu where maas in(select max(maas) from PersTablosu group by bolNo)
```

- **1. bölüm dışındaki bölümlerde en yüksek maaş alanlar kimler?**

```
select ad,maas,bolNo from PersTablosu where maas in(select max(maas) from PersTablosu group by bolNo
having bolNo>1)
```

- **En yüksek maaşı alan personelin adı maaşı ve yöneticisinin ismi?**

select ad,yonadisoyadi from PersTablosu,Yonetici where maas=(select max(maas) from PersTablosu) and ySosGuvNo=yonSosGno

- **En düşük maaşı alan personelin yöneticisinin bölümü?**
- select bolAd,yonadisoyadi,ad from PersTablosu,Yonetici,BolumTablosu

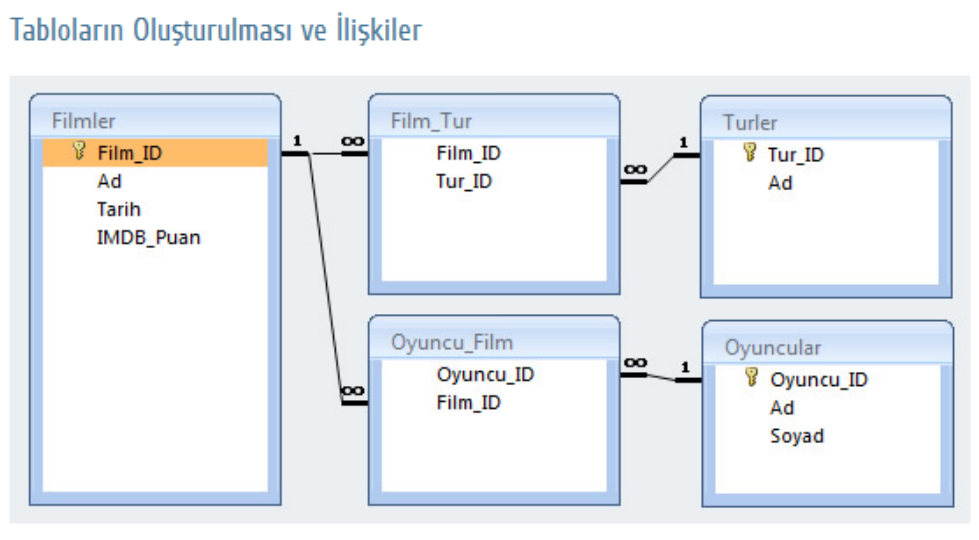
  where maas=(select min(maas) from PersTablosu) and

  PersTablosu.ySosGuvNo=Yonetici.yonSosGno and Yonetici.yonSosGno=BolumTablosu.yonSosGno

by yselim

SQL – İç İçe Select-Örnek uygulama

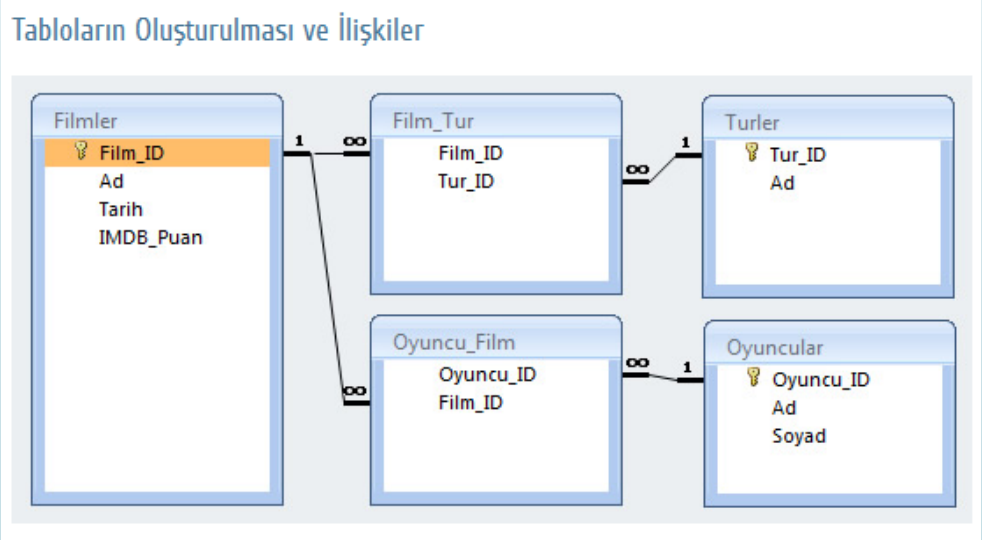
- Örnek başka bir uygulama ile iç içe select konusunu pekiştirelim.
- Aşağıdaki yapıda bir film oyuncu veritabanı oluşturacağız. Aralarındaki ilişkileri aşağıdaki şekilde ayarlayıp iç içe select örnekleri yapalım.
- Veritabanını online içerikten indirebilirsiniz.



by yselim

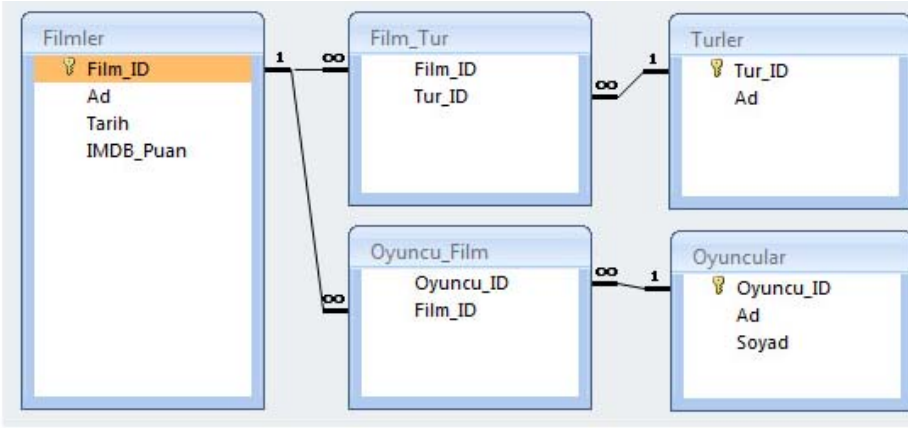
SQL – İç İçe Select-Örnek uygulama

- Filmler ile Oyuncular arasında **sonsuza sonsuz ilişki** kurup, **gereksiz veri tekrarı** yapmayı önlemek amacıyla araya bir **geçiş tablosu** ekledik. Bu şekilde veri tekrarı **minimize** edildi. Aynı şeyi Filmler ile Türler arasında da yapıldı
- İlişkileri bu şekilde kurduktan sonra, alt sorgu kavramına dönüp, devam edelim.



by yselim

SQL – İç İçe Select-Ö



Film_No	Ad	Tarih	IMDB_Puan	Tur_ID
1	The Shawshank Redemption	1994	9,2	1
2	The Dark Knight	2008	8,8	7
3	Fight Club	1999	8,8	1
4	Forrest Gump	1994	8,6	
5	Leon	1994	8,6	1
6	V For Vandetta	2006	8,1	7
7	A Beautiful Mind	2006	8	
8	Eternal Sunshine of the Spotless Mind	2004	8,4	2
9	Inception	2010	8,8	3
10	Good Will Hunting	1997	8,1	1

	oyuncuID	Ad	Soyad
1	1	tom	hanks
2	2	natalia	portman
3	3	jim	carrey
4	4	max	style

Oyuncular

	filmId	turid
1	1	1
2	2	7
3	3	1
4	4	NULL
5	5	4
6	6	2
7	7	NULL
8	8	3

Film-tur

	oyuncuId	filmId
1	1	1
2	2	2
3	1	4
4	3	5
5	4	6
6	4	7
7	3	3
8	2	8

Oyuncu-film

Tur_ID	Ad
1	Dram
2	Romantik
1	Dram
3	Bilim Kurgu
4	Macera
1	Dram
5	Gerilim
6	Korku
7	Suç

by yselim

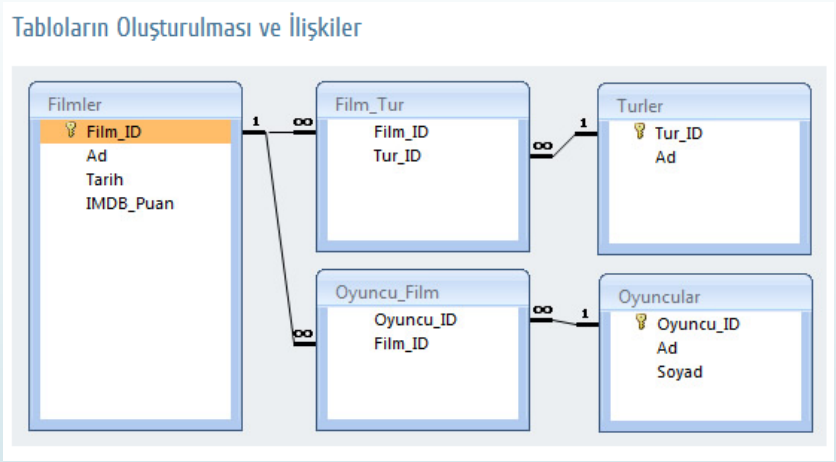
SQL – İç İçe Select-Örnek uygulama

	filmId	turid
1	1	1
2	2	7
3	3	1
4	4	NULL
5	5	4
6	6	2
7	7	NULL
8	8	3

Film-tur

	oyuncuId	filmId
1	1	1
2	2	2
3	1	4
4	3	5
5	4	6
6	4	7
7	3	3
8	2	8

Oyuncu-film



by yselim

SQL – İç İçe Select

- Tom Hanks'in oynadığı filmleri getirsin ;
- Türü macera olan filmlerin Adlarını ve IMDB Puanlarını listeleyelim
- Natalia Portman'ın oynadığı film sayısını bulalım ;
- IMDB Puanı, tüm filmlerin IMDB Puanlarının ortalamasından büyük olan dram filmlerin adlarını, IMDB puanlarını ve tarihlerini listeleyelim
- Tom Hanks'in oynadığı dramatik filmlere bakalım ;

by yselim

SQL – İç İçe Select

Örnek bir alt sorgu yapalım ve bize Tom Hanks'in oynadığı filmleri getirsin ;

```
SELECT Ad FROM Filmler WHERE Film_ID IN ( SELECT
Film_ID FROM Oyuncu_Film WHERE Oyuncu_ID = ( SELECT
Oyuncu_ID FROM Oyuncular WHERE Ad = "Tom" AND Soyad
= "Hanks" ) )
```

SQL – İç İçe Select

Aşağıda iç içe sorgular ile bilinmesi gereken noktalar var ;

Alt sorgu, bir üstteki sorguda yer alan **WHERE**, **HAVING** veya **FROM** kelimeleri içine yazılırlar. Yukarıda WHERE içinde bir alt sorgu ve onun altındaki WHERE içine de bir başka alt sorgu yazılmıştı.

Alt sorgular parantez işaretleri içerisinde yazılmalıdır.

Alt sorgularda **ORDER BY** komutu kullanılamaz, ancak ana sorguda kullanabilirsiniz.

Tek Satır Döndüren Alt Sorgular : Kriter belirteçleri =, <, > , <=, >= veya <> olabilir.
Çok Satır Döndüren Alt Sorgular : Kriter belirteçleri IN, ANY veya ALL olabilir.

SQL – İç İçe Select

Türü dram olan filmlerin Adlarını ve IMDB Puanlarını listeleyelim ;

```
SELECT Ad, IMDB_Puan FROM Filmler WHERE Film_ID IN ( SELECT
Film_ID FROM Film_Tur WHERE Tur_ID IN ( SELECT Tur_ID FROM
Turler WHERE Ad = "Dram" ) )
```

Açıklama : En alttaki sorguya bakılırsa, türü dram olan filmlerin Tur_ID'leri bir üstteki sorguya gönderilmiştir. Ortadaki sorgu ifadesi ise Tur_ID'si dram olan filmlerin Film_ID'lerini ana sorguya göndermiştir. Bütün kriterler bu iki alt sorgu ile sağlandıktan sonra en üstteki SELECT ifadesi ile Ad ve IMDB_Puan değerleri seçilerek listelenmiştir.

SQL – İç İçe Select

Natalia Portman'ın oynadığı film sayısını bulalım ;

```
SELECT COUNT(Film_ID) AS Film_Sayisi FROM Filmler WHERE
Film_ID IN ( SELECT Film_ID FROM Oyuncu_Film WHERE Oyuncu_ID
IN ( SELECT Oyuncu_ID FROM Oyuncular WHERE Ad = "Natalia" AND
Soyad = "Portman" ) )
```

Açıklama : En alttaki sorguda adı Natalia soyadı Portman olan oyuncunun Oyuncu_ID'si bir üstteki sorguya gönderilmiştir. Ortadaki sorgu ise, alt sorgudan gelen Oyuncu_ID değerini barındıran filmlerin Film_ID'lerini bir üstteki sorguya göndermiştir. (Dikkat ederseniz Oyuncu_Film tablosunda 2 sütun var : Oyuncu_ID, Film_ID) Ana sorguda ise alt sorgulardaki kriterlerden geçen kayıtların sayısı, yani Natalia Portman'ın oynadığı filmlerin sayısı bulunmuştur.

by yselim

SQL – İç İçe Select

IMDB Puanı, tüm filmlerin IMDB Puanlarının ortalamasından büyük olan dram filmlerin adlarını, IMDB puanlarını ve tarihlerini listeleyelim ;

```
SELECT Ad, Tarih, IMDB_Puan FROM Filmler WHERE IMDB_Puan > (
SELECT AVG(IMDB_Puan) FROM Filmler ) AND Film_ID IN ( SELECT
Film_ID FROM Film_Tur WHERE Tur_ID IN ( SELECT Tur_ID FROM
Turler WHERE Ad = "Dram" ) )
```

Açıklama : Bu sorguda 2 farlı kriter mevcuttur : IMDB puanı, ortalamanın üstünde olan filmler ve dram filmleri.
Dramatik filmleri bulmak amacıyla en alttaki sorgudan, türü dram olan filmlerin Tur_ID'leri bir üsttekine gönderilmiştir. Bir üstteki sorguda ise, dramatik filmlerin Film_ID'lerini ana sorguya göndermiştir.
Ortalama puanların üzerindeki filmler bulmak amacıyla AVG fonksiyonu kullanarak ana sorguya bir kriter daha belirlemiş olduk. 2 kriterde sağlandığına göre artık Ad, Tarih ve IMDB_Puan değerlerini SELECT edebiliriz.

by yselim

SQL – İç İçe Select

Tom Hanks'in oynadığı dramatik filmlere bakalım ;

```
SELECT Ad FROM Filmler WHERE Film_ID IN ( SELECT Film_ID FROM
Oyuncu_Film WHERE Oyuncu_ID = ( SELECT Oyuncu_ID FROM
Oyuncular WHERE Ad LIKE "Tom*" ) ) AND Film_ID IN ( SELECT
Film_ID FROM Film_Tur WHERE Tur_ID = ( SELECT Tur_ID FROM
Turler WHERE Ad LIKE "Dra*" ))
```

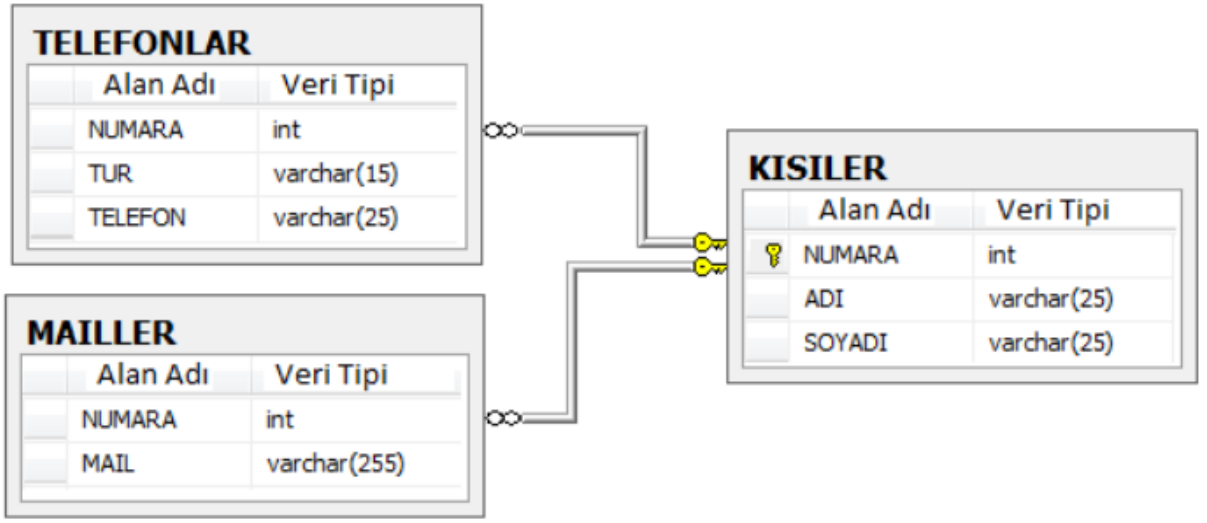
Açıklama : Bu sorgu için 5 tabloyu da kullanmamız gerekiyor. Yine 2 farklı kritere uyan filmleri listeleyeceğiz.
Önce Tom Hanks'in oynadığı filmlerin Film_ID'leri, daha sonra türü dram olan filmlerin Film_ID'leri ana sorguya gönderilmiştir. Bu 2 kriteri sağladıktan sonra SELECT ile listeleme yapabiliriz.

SQL – JOIN kullanımı

- JOIN (birleştir, eşleştir) bir veri tabanında iki veya daha fazla veri tabanı tablosunu ortak bir paydada bir araya getirmek amacıyla kullanılan SQL sorgu işlemidir.
- Örneğin bir tablonuzda Kişiler var, diğer iki tablomuzda ise bu kişilerin telefonları ve mail adresleri olmuş olsun. Bazı kişilerin telefon veya mail adreslerini sorgulamak istediğimizde önce kişinin numarasını daha sonrada bu numaraya göre kişinin mail veya telefon numarasını sorgulayabiliriz.
- Bunun için en az iki veya üç sorgu yazmamız gerekecektir. Böyle durumlarda JOIN işlemi bunu tek sorguda yapabilmenizi sağlar.
- Tablo birleştirirken, WHERE ile birleştirme koşulu kullanmak yerine **JOIN** ifadelerini kullanabilirsiniz. Bu ifadeler dörde ayrılı ve şu şekildedir ;
- **INNER JOIN (where)**
- **LEFT JOIN (soldan bağlama)**
- **RIGHT JOIN (sağdan bağlama)**
- **FULL JOIN (sol ve sağ bağlama)**

SQL – JOIN kullanımı

Aşağıdaki örnek veri tabanının tabloları arasındaki ilişkilere dikkat edelim. **JOIN** işlemlerini bu tablolara göre yapacağız.



Bu tablolara göre bir kişinin sonsuz telefon numarası veya sonsuz mail adresi olabilmektedir. Örneğin kişilere ait telefon numaralarını sorgulamak istersek

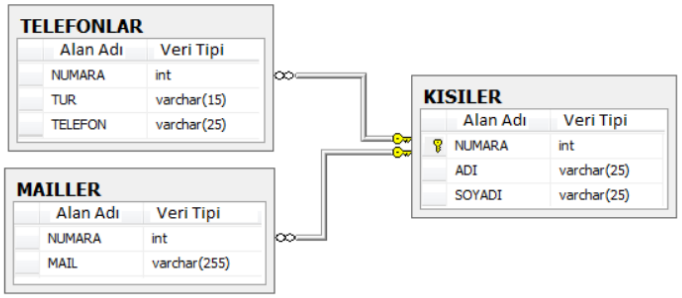
```
Select Adı,Telefon from kisiler,telefonlar
where kisiler.numara=telefonlar.numara
```

Bu sorgulama sonucu ile kişilere ait varsa bütün telefon numaraları gelecektir. Fakat telefon numarası olmayan kişi kayıtları gelmeyecektir.

by yselim

SQL – INNER JOIN kullanımı

Aşağıdaki örnek veri tabanının tabloları arasındaki ilişkilere dikkat edelim. JOIN işlemlerini bu tablolara göre yapacağız.




```
Select Adı,Telefon from kisiler,telefonlar
where kisiler.numara=telefonlar.numara
```

- Yukaridaki sorgunun aynısını **INNER JOIN** komutu ile aşağıdaki gibi yazılabilir

```
SELECT K.NUMARA,ADI,SOYADI,TELEFON FROM KISILER K INNER JOIN
TELEFONLAR T ON K.NUMARA =T.NUMARA
```

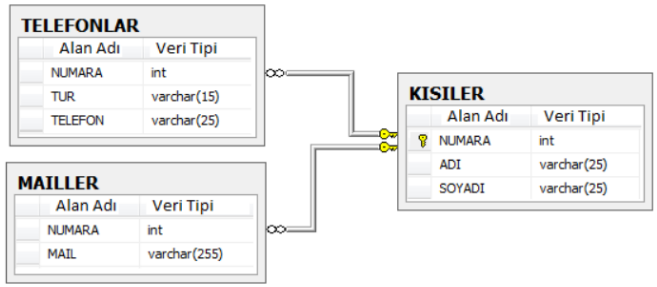
Bu sorgunun sonucunda sadece numarası eşleşen kayıtlar gelecektir. Yani where komutu gibi çalışır.

	NUMARA	ADI	SOYADI	MAIL
1	1	ALİ	KARA	alikara@gmail.com
2	6	Selçuk	BİLSİN	selcukbilsin@hotmail.com
3	6	Selçuk	BİLSİN	selcukkk@gmail.com
4	7	Faruk	KIRMIZI	farukkirmizi@gmail.com
5	7	Faruk	KIRMIZI	farukkirmizi@msn.com
6	5	Ebru	ÇAPKIN	ebrucapkin@tmail.com
7	3	Özge	ÇITAK	ozgecitak@mmail.com

by yselim

SQL – LEFT (OUTER) JOIN kullanımı

Aşağıdaki örnek veri tabanının tabloları arasındaki ilişkilere dikkat edelim. JOIN işlemlerini bu tablolara göre yapacağız.



Eğer mail tablosunda olmayan kişilerin de sorguda görünmesi isteniyorsa bu durumda aşağıdaki gibi bir sorgu yazılabilir.

```
SELECT K.NUMARA,ADI,SOYADI,MAIL FROM KISILER K LEFT OUTER JOIN MAILLER M
```

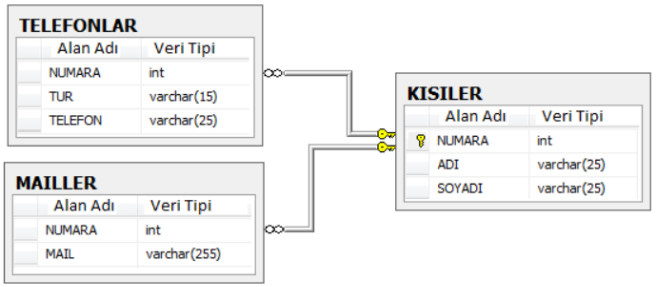
	NUMARA	ADI	SOYADI	MAIL
1	1	ALİ	KARA	alikara@gmail.com
2	2	AYŞE	BEYAZ	NULL
3	3	Özge	ÇITAK	ozgecitak@mmail.com
4	4	Buse	BİLGİÇ	NULL
5	5	Ebru	ÇAPKIN	ebrucapkin@tmail.com
6	6	Selçuk	BİLSİN	selcukbilsin@hotmail.com
7	6	Selçuk	BİLSİN	selcukkk@gmail.com
8	7	Faruk	KIRMIZI	farukkirmizi@gmail.com
9	7	Faruk	KIRMIZI	farukkirmizi@msn.com
10	8	Mesut	KARA	NULL
11	9	Haydar	YAŞİN	NULL
12	10	Hamdi	KARBONAT	NULL

- Bu sorgulama sonucuna dikkat edilirse sol tarafa yazılan tablo esas tablo olarak kabul edilmektedir.
- Bu sorgulama ile sol tablodaki tüm kayıtlar getirilir. Buna karşılık sağ tarafa yazılan tablodan kayıtlar getirilerek sol taraftaki kayıtlar ile eşleştirilir.
- Eşleştirme bu sorgulamada
  ON K.NUMARA =M.NUMARA
  şeklinde yapılmıştır.
- Eğer sol tablodaki kayda karşılık sağ tarafta eşleşen kayıt yoksa karşısı NULL olarak kalmaktadır

by yselim

SQL – LEFT (OUTER) JOIN kullanımı

Aşağıdaki örnek veri tabanının tabloları arasındaki ilişkilere dikkat edelim. JOIN işlemlerini bu tablolara göre yapacağız.



- Bir başka örnek olarak *telefonlar* tablosu ile *mailler* tablosunu birleştirin??

```
SELECT   T.NUMARA, TUR, TELEFON, MAIL
FROM     TELEFONLAR T LEFT OUTER JOIN MAILLER M
         ON T.NUMARA = M.NUMARA
```

	NUMARA	TUR	TELEFON	MAIL
1	1	Ev	0534-324-36-42	alikara@gmail.com
2	1	İş	0535-324-24-45	alikara@gmail.com
3	2	Cep	0536-324-58-79	NULL
4	2	Ev	0537-524-67-45	NULL
5	3	Ev	0544-624-34-87	ozgecitak@mmail.com
6	4	Ev	0554-724-34-45	NULL
7	5	Fax	0564-824-34-45	ebrucapkin@tmail.com
8	5	İş	0544-394-34-45	ebrucapkin@tmail.com
9	5	Ev	0524-024-34-78	ebrucapkin@tmail.com
10	6	Cep	0554-524-34-45	selcukbilsin@hotmail.com
11	6	Cep	0554-524-34-45	selcukkk@gmail.com

LEFT JOIN birleştirmesinde ana tablo sol tarafta yazılan tablo olduğundan sol taraftaki telefonlar tablosundaki tüm kayıtlar getirilmiş, karşılığında varsa mailler tablosundan kayıtlar getirilmiştir. Karşılığı yoksa NULL bırakılmıştır.

SQL – RIGHT (OUTER) JOIN ve FULL (OUTER) JOIN kullanımı

RIGHT JOIN; LEFT JOIN birleştirmesinin tam tersi olarak davranmaktadır. RIGHT JOIN birleştirmesinde ana tablo sağdaki tablo olmaktadır.

FULL OUTER JOIN birleştirmelerinde ise her iki tabloda da olan tüm kayıtlar birleştirilir. Varsa eşleşen kayıtlar eşleştirilir. Yoksa karşıları NULL olarak kalır.

```
SELECT    T.NUMARA, TUR, TELEFON, MAIL
FROM      TELEFONLAR T FULL OUTER JOIN MAILLER M
          ON T.NUMARA = M.NUMARA
```

Bu sorgulama ile hem telefonlar tablosundaki tüm kayıtlar hem de mailler tablosundaki tüm kayıtlar eşleştirilerek görüntülenmiştir. Telefonu olup maili olmayan veya maili olup telefonu olmayan kayıtların karşılıkları NULL olarak bırakılmıştır.

	NUMARA	TUR	TELEFON	MAIL
1	1	Ev	0534-324-36-42	alikara@gmail.com
2	1	İş	0535-324-24-45	alikara@gmail.com
3	2	Cep	0536-324-58-79	NULL
4	2	Ev	0537-524-67-45	NULL
5	3	Ev	0544-624-34-87	ozgecitak@mmail.com
6	4	Ev	0554-724-34-45	NULL
7	5	Fax	0564-824-34-45	ebrucapkin@tmail.com
8	5	İş	0544-394-34-45	ebrucapkin@tmail.com
9	5	Ev	0524-024-34-78	ebrucapkin@tmail.com
10	6	Cep	0554-524-34-45	selcukbilsin@hotmail.com
11	6	Cep	0554-524-34-45	selcukkk@gmail.com
12	NULL	NULL	NULL	farukkirmizi@gmail.com
13	NULL	NULL	NULL	farukkirmizi@msn.com

by yselim

SQL – JOIN Uygulama (Pers vt için)

```
--Boş personel kaydı ekle. sadece sicil alanı var
-- boş kaydı olan yönetici kaydı, yonsosgno var sadece
-- boş kaydı olan bölüm, sadece bolumno var.
use pers
--inner join
select p.sicilNo,p.ad,p.soyad,yonSosGno from PersTablosu as p,Yonetici y where
y.yonSosGno=p.ySosGuvNo
select p.sicilno,p.ad,p.soyad,yonSosGno from PersTablosu as p  inner join yonetici y on
y.yonSosGno=p.ySosGuvNo
--left join
select p.ad,p.soyad,p.sicilNo,y.yonAdiSoyadi from PersTablosu as p left outer join yonetici y on
y.yonSosGno=p.ySosGuvNo
--right join full join
select Yonetici.yonAdiSoyadi,BolumTablosu.bolAd from BolumTablosu,Yonetici where
Yonetici.yonSosGno=BolumTablosu.yonSosGno
select y.yonAdiSoyadi,y.yonSosGno,b.bolAd,b.bolumNo from Yonetici y join BolumTablosu b on
y.yonSosGno=b.yonSosGno
select y.yonAdiSoyadi,y.yonSosGno,b.bolAd,b.bolumNo from Yonetici y right outer join BolumTablosu b
on y.yonSosGno=b.yonSosGno
select y.yonAdiSoyadi,y.yonSosGno,b.bolAd,b.bolumNo from Yonetici y left outer join BolumTablosu b
on y.yonSosGno=b.yonSosGno
select y.yonAdiSoyadi,y.yonSosGno,b.bolAd,b.bolumNo from Yonetici y full outer join BolumTablosu b
on y.yonSosGno=b.yonSosGno
select b.bolAd,b.yonSosGno,b.bolumNo from BolumTablosu b right outer join Yonetici y on
y.yonSosGno=b.yonSosGno
```

by yselim

SQL – JOIN kullanımı

- ÖZET:
- Tablo birleştirme işlemini 2 farklı şekilde yapabilirsiniz ; WHERE ile sorgu koşulu kullanmak ve JOIN ifadelerini kullanmak.
- WHERE ile sorgu koşulu oluşturmak için 2 tablonun Primary Key ve Foreign Key sütunları baz alınır ve onlar üzerinden birleştirme işlemi yapılır.
- JOIN ifadeleri ile tablo birleştirilirken eşleşme durumları önemlidir. Eşleşme durumalarına göre ayrılmaktadırlar : INNER, LEFT, RIGHT,FULL.

by yselim

SQL - EXIST Komutu

- SQL dilinde bir sorgunun içinde kayıt olma durumunu belirlemek amacıyla EXIST komutu kullanılır,

- SELECT * FROM tablo_adi WHERE EXISTS (SELECT * FROM tablo_adi2 WHERE tabloadi.alan1=tabloadi.alan2);

by yselim

SQL - EXIST Komutu

- Egitselyazilim veri tabanındaki 1 no'lu içeriği görüntüleyen kullanıcıların ad ve soyadlarını görüntülemek için;

- SELECT ad,soyad FROM kullanici_bilgileri WHERE EXISTS (SELECT * FROM gezinme WHERE kullanici_bilgileri.ogrno=gezinme.ogrno and gezinme.icerikno=1);

SQL – NOT EXIST Komutu

- SQL dilinde bir sorgunun içinde bulunmayan kayıtları belirlemek amacıyla NOT EXIST komutu kullanılır,

- SELECT * FROM tablo_adi WHERE NOT EXISTS (SELECT * FROM tablo_adi2 WHERE tabloadi.alan1=tabloadi2.alan2);

by yselim

SQL – NOT EXIST Komutu

- Egitselyazilim veri tabanındaki 1 no'lu içeriği görüntülemeyen kullanıcıların ad ve soyadlarını görüntülemek için;

- SELECT ad,soyad FROM kullanici_bilgileri WHERE NOT EXISTS (SELECT * FROM gezinme WHERE kullanici_bilgileri.ogrno=gezinme.ogrno and gezinme.icerikno=1);

by yselim