* Örneğin notlar tablosunda sadece notu alan ogrenci numarası ve not aldığı tarihi listelemek için;
* SELECT ogrno,tarih FROM notlar;
* Örneğin alınan notları tarihe göre sıralamak için;
* SELECT \* FROM notlar ORDER BY tarih;
* Örneğin alınan notları tarihe göre azalan sırada sıralamak için;
* SELECT \* FROM notlar ORDER BY tarih DESC;
* Örneğin alınan notları alındığı içerik numaralarına göre azalan, öğrenci numaralarına göre de artan sırada listelemek için;
* SELECT \* FROM notlar ORDER BY icerikno DESC, ogrno ASC ;
* Örneğin sadece 9801 no’lu öğrencinin aldığı notları görüntülemek için;
* SELECT \* FROM notlar WHERE ogrno=9801;
* Örneğin öğrenci numarası 9801 ya da 9802 olan 1 nolu içeriği önemli sayfa yapan kullanıcılar varsa listelemek için;
* SELECT \* FROM onemli\_sayfalar WHERE (ogrno=9801OR ogrno=9802) AND icerikno=1;
* Eğitselyazılım veri tabanındaki adı Sema veya Serdar olan kullanıcıları görüntülemek için;
* SELECT \* FROM kullanici\_bilgileri WHERE ad IN (‘Sema’ ,’Serdar’);
* 1. soruya verilen puanların toplamını bulmak için;
* SELECT SUM(puan) FROM tartisma\_soru\_puanlar where soruno=1;
* Örneğin 1 no’lu tartışma sorusuna verilen puanların ortalaması yuvarlayarak görüntülemek için;
* SELECT ROUND(AVG(puan)) FROM tartisma\_soru\_puanlar where soruno=1;
* Örneğin her bir konuyu kaç kişinin önemli sayfa yaptığını belirlemek için;
* SELECT icerikno , COUNT (\*) as toplamnot from notlar GROUP BY icerikno;
* SELECT DISTINCT yasi, bolumu FROM businif komutu ne işe yarar.
* Bu sinif tablosunda yaşı aynı olan kişilerden sadece bir tanesini ve bölümünü listeler.
* Doğum tarihi 1960’dan önce olan maaşı 6000000 – 10000000 arasındaki bayan personeli listeleyiniz.
* SELECT \* FROM personel WHERE dog\_tar < {01/01/60} AND brüt >= 6000000 AND brüt <= 10000000 AND cinsiyet = 'Bayan‘
* Bölümü Satış ya da Muhasebe olamayan 1960’dan sonra doğmuş bayan personeli listeleyiniz.
* SELECT \* FROM personel WHERE NOT (böl\_no =1 OR böl\_no =2) AND dog\_tar >={01/01/60} AND cinsiyet = 'Bayan'

Doğum yeri Artvin, Rize ve Trabzon olmayan kayıtları listelemek için : Select \* from personel Where dogumyeri NOT IN ('Artvin', 'Rize', 'Trabzon')

* Görev 10’da oluşturulan veri tabanındaki adı S ile başlayan kullanıcıların kimler olduğunu görüntülemek için;
* SELECT \* FROM kullanici\_bilgileri WHERE ad LIKE 'S%';
* Görev 10’da oluşturulan veri tabanındaki 01.03.2009 ile 03.03.2009 tarihleri arasında not alan öğrencileri ve aldıkları notları listelemek için;
* SELECT ogrno, notu FROM notlar WHERE tarih BETWEEN ‘03/01/2009’ AND ‘03/03/2009’;
* SELECT Count(\*) FROM arayuz\_tercihleri WHERE id IN (SELECT DISTINCT arayuzno FROM arayuz\_tercihleri);
* Oluşturulan veri tabanında kaç farklı arayüz seçimi yapıldığını belirleyin.
* Örneğin 3 öğrenci 3,3 ve 5 no’lu arayüzü seçtiklerinde seçilen farklı arayüz sayısı 2’dir.
* İsminde ikinci harfi A dördüncü harfi B olan öğrencilerin bilgilerini veren sorguyu yazınız.
* SELECT \* FROM ogrenci WHERE adi LIKE '\_A\_B%'
* **Örnek:** İsminde AL geçen öğrencilerin bilgilerini listeleyen sorguyu yazınız.
* SELECT \* FROM ogrenci WHERE adi LIKE '%AL%‘
* **Örnek:**Adının başında Fırat geçen öğrencilerin bilgilerini veren sorguyu yazınız.
* SELECT \* FROM ogrenci WHERE adi LIKE 'Firat%‘
* **Örnek:** Adının sonunda Fırat geçen öğrencilerin bilgilerini bulan sorguyu yazınız.
* SELECT \* FROM ogrenci WHERE adi LIKE '%Firat‘
* **Örnek:**Adının ikinci harfi A olan ve uzunluğu belli olmayan öğrencinin bilgilerini veren sorguyu yazınız.
* SELECT \* FROM businif WHERE adi LIKE '\_A%‘
* **Örnek:** SELECT \* FROM personel WHERE adres LIKE ‘%TAKSİM%’ sorgusunun işlevini yazınız.
* Adres LIKE ‘%TAKSİM%’ ifadesi adres içinde her hangi bir yerde TAKSİM yazan yerde oturan personeli listeleyecektir
* Bolkod=531 olan öğrencilerin not bilgilerini veren sorguyu yazınız.
* SELECT \* FROM notlar WHERE numara IN ( SELECT numara FROM ogrenci WHERE bolkod=531)
* **Örnek:** SELECT adi, soyadi, no, adres FROM ogrenci WHERE no IN (1,3,5,7) sorgusunun işlevini yazınız.
* **Örnek:**123 nolu öğrencinin 3 kredilik derslerden aldığı notları listeleyen sorguyu yazınız.
* SELECT vize,final FROM notlar WHERE ogr\_no= 123 AND bolkod IN( SELECT bolkod FROM dersler WHERE kredi=3)
* Bölüm kodu 1 ve 4 olan derslerden notları açıklanan öğrencilerin numarasını bulan sorguyu yazınız.
* select distinct(ogr\_no) from notlar where bolkod in(1,4)
* Bölüm kodu 1 ve 4 olan derslerden notları açıklanan öğrencilerin numarasını bulan sorguyu yazınız.
* select distinct(ogr\_no) from notlar where bolkod in(1,4)
* bölüm kodu 1 ve 3 olan öğrencilerin adını ve soyadını veren sorguyu yazınız.
* select adi,soyadi from ogrenci where bolkod in(1,3)
* stok tablosunda en pahalı malın bilgilerini veren SQL komutu nasıl olmalıdır ? SELECT TOP 1 \* FROM stok ORDER BY fiyat DESC
* stokta en az hangi maldan kalmıştır ? SELECT TOP 1 \* FROM stok ORDER BY adet ASC
* stokta ortalama fiyattan pahalı olan malların listesi ? SELECT \* FROM stok WHERE fiyat > ( AVG(fiyat) ) -- YANLIŞ SELECT \* FROM stok WHERE fiyat > ( select AVG(fiyat) from stok )
* SELECT \* FROM musteri WHERE adres LIKE ‘%BURDUR%’ --Adres alanının herhangi bir yerinde ‘BURDUR’ kelimesi geçen kayıtları listeler SELECT \* FROM musteri WHERE ad LIKE ‘\*AKR+\_ \_ \_ \_’ --ad alanının ilk karakteri A veya K olan ve toplam 5 karakter olan kayıtları listeler SELECT \* FROM musteri WHERE ad LIKE ‘\*^MCS+ %’ --ad alanının ilk karakteri M veya C veya S ile başlamayan kayıtları listeler
* SELECT \* FROM stok WHERE stokadi IS NULL --stok adi girilmeyenleri listeler SELECT \* FROM stok WHERE stokadi IS NOT NULL --stok adi girilenleri listeler
* Soru : stok tablosunun kalan malları içerecek bir kopyasını oluşturunuz ? SELECT \* into stok\_kopya FROM stok WHERE adet>=1
* Soru : satışı yapılan malların listesi ? SELECT \* FROM stok WHERE sno NOT IN( select DISTINCT sno from satis)
* Fatih’te oturan personel çalıştığı projeler adları ve yerleri listeleyin.
* SELECT proje\_ad,yer FROM Proje WHERE bl\_no in(SELECT bol\_no FROM Personel WHERE adres LIKE '%Fatih%');
* Soru : son kullanma tarihine 20 gün veya daha az kalmış malların listesi ? SELECT \* FROM stok WHERE sonkultar <= ( DateAdd(day,20,GetDate() ) )
* Çalışma saati 250 olan projelerin yer bilgisini listeleyin.
* SELECT yer FROM Proje WHERE proj\_no in(SELECT proje\_no FROM Calisma WHERE saat=250);
* Ağırlığı 500 olan parçaların kullanıldığı bir projenin çalışma saatlerini listeleyiniz.
* SELECT saat FROM Calisma where proje\_no in (SELECT pr\_no FROM Parca where agirlik=500)
* Adı Metin Yılmaz olan öğrencinin hangi derslerden sınava girdiğini ve sınav sonuçlarını listeleyiniz.
* SELECT ders\_adi,vize,final FROM notlar where ogr\_no in(SELECT ogr\_no FROM ogrenci where adsoyad='metin yılmaz')
* Finalden en düşük notu alan öğrencinin adının bulalım
* select adsoyad from ogrenci where ogr\_no =(select ogr\_no from notlar where final=(select min(final) from notlar))
* Hangi derslerden büte kalınmış onu bulalım
* select avg(yas) as ortalama from ogrenci where ogr\_no in (select ogr\_no from notlar where ((vize\*0.4)+(final\*0.6))<50)
* Aşağıdaki SELECT komutu ile bir Calisan tablosunda personelin şu anda geçerli olan maaşı ile bu maaşın %10 zamlı şekli listelenmektedir.
* SELECT ad, soyad, maas, maas\*1.1 as zamli, maas\*1.1-maas as zam FROM Calisan;
* SELECT adsoyad, yas\*2 FROM ogrenci WHERE yas<14 Komutunun işlevini yazınız.
* Yaşı 14’den küçük olanların adını ve yaşının 2 ile çarpımını verir.
* SELECT SUM(maas\*0.1) FROM Calisan;
* Sorgusunun işlevini yazınız.
* Maaşların %10’larını toplar ve listeler.
* Tüm hocaların girdiği toplam ders saatini bulan sorguyu yazınız. (Okul Projesi)
* select SUM(top\_ders\_saati)as toplam from hocalar
* Tüm sınavların ortalamasını veren sorguyu yazınız.
* select AVG(vize),AVG(final)from notlar
* SELECT COUNT(\*) FROM ogrenci; Sorgusunun çıktısı ne olur.
* Toplam öğrenci sayısını verir.
* select COUNT(\*) from ogrenci where cinsiyet='bayan' Çıktısını bulunuz.
* Öğrenci tablosunda bulunan toplam bayan sayısını verir.
* Kaç farklı isme sahip öğrenci sayısını veren sorguyu bulunuz.
* SELECT COUNT(DISTINCT adi) FROM ogrenci;
* Notlar tablosuna kaç farklı dersin not girildiğini bulan sorguyu ve çıktıyı yaznız.
* select COUNT(distinct(ders\_adi)) from notlar
* Final notu 60’ın altında olan ogrenci sayısı nedir?
* SELECT COUNT(final) FROM notlar WHERE final<60
* Bölüm numarası 2 olan personellerden her hangi birinden daha düşük maaş alan kişileri listeleyiniz.
* SELECT \* FROM personel WHERE brut<ANY( SELECT brut FROM personel WHERE bol\_no=2 )
* Bölüm kodu 1 olan dersin vize notundan herhangi birinden vize notu yüksek olan öğrencilerin numaralarını veren sorguyu yazınız.
* select ogr\_no from notlar where vize>any(select vize from notlar where bolkod=1)
* Final notu 65 ten küçük olan final notlarının herhangi birinden final notu düşük olanların final ortalamasını veren sorguyu yazınız.
* select avg(final) as ortalama from notlar where final<any(select final from notlar where final<65 )
* : Bölüm kodu 1 olan personellerin brüt maaşından fazla maaşı olan personellerin bilgilerini 1
* select \* from personel where brut>all(select brut from personel where bol\_no=1)
* ya da
* select \* from personel where brut>(select max(brut) from personel where bol\_no=1)
* Bölüm kodu 2 olan final notlarının hepsinden final notu büyük olanların not bilgilerini listeleyen sorguyu yazınız.
* select \* from notlar where final>all(select final from notlar where bolkod=2)
* Adı Nermin ince olan öğrencinin final notunu bulan sorguyu yazınız.
* select final from notlar where ogr\_no in(select ogr\_no from ogrenci where adsoyad='nermin ince')
* 50’den aşağı not alan öğreciler bütünlemeye kalacaktır. Bütünlemeye kalacak öğrencilerin ad ve soyadlarını listeleyen sorguyu yazınız.
* select adsoyad from ogrenci where ogr\_no in (select ogr\_no from notlar where ((vize\*0.4)+(final\*0.6))<50)
* Finalden en düşük notu alan öğrencinin adının bulan sorguyu yazınız.
* select adsoyad from ogrenci where ogr\_no=(select ogr\_no from notlar where final=(select min(final) from notlar))
* Parça numarası 24 olan parçayı kullanan projelerde çalışan personeli listeleyiniz. Örnek olarak aşağıdaki verileri göz önüne alalım.
* SELECT \* FROM Personel WHERE sosyal\_g\_no in(SELECT Per\_s\_g\_no FROM Calisma WHERE proje\_no in(select proj\_no from Proje where proj\_no in(select pr\_no from Parca where par\_no=24)));
* Egitselyazilim veri tabanındaki Ayşe isimli kullanıcının kullanıcı adını değiştirmek için;
* UPDATE kullanici SET kul\_adi=‘ayse\_bote’ WHERE adi=‘Ayşe’;
* Egitselyazilim veri tabanındaki kullanıcı tablosundaki yer alanının veri türüyle ilgili değişiklik yapmak için;
* ALTER TABLE kullanici ALTER COLUMN yer VARCHAR(100);
* Egitselyazilim veri tabanındaki kullanıcı tablosundaki yer alanı kaldırmak için;
* ALTER TABLE kullanici DROP COLUMN yer;
* Verilen notlar tablosunda 1.sınava veya 2.sınava girmeyen öğrencilerin numarasını ve sınava girmediği dersin kodunu bulmak için gerekli SQL ifadesini yazınız.
* SELECT ogr\_no,ders\_kodu FROM ogrenci\_yeni WHERE sinav1 IS NULL OR sinav2 IS NULL
* Birinci sınava girmeyen öğrencilerin isimlerini bulan sql sorgusunu yazınız.
* 1
* Bütünleme sınavına giren öğrencilerin adlarını, hangi dersten bütünleme sınavına girdiklerini ve ders adlarına göre gruplandıran sorguyu yazınız.
* select adi,ders\_adi from ogrenci,dersler where no in(select no from notlar where but is not null and op\_kod in(select op\_kod from dersler)) group by ders\_adi,adi
* DELETE FROM ogrenci; sorgusunun işlevini yazınız.
* Tablodaki tüm kayıtları siler.
* **Örnek:** DELETE FROM ogrenci WHERE adi = ‘necmi büyük'; sorgusunun işlevini yazınız.
* Adı necmi büyük olan tüm kayıtları siler.
* **Örnek:** DELETE FROM ogrenci WHERE og\_no=130; sorgusunun işlevini yazınız.
* Numarası 130 olan kaydı siler.
* DELETE FROM ogrenci WHERE yas<13; sorgusunun işlevini yazınız.
* Yaşı 13’den küçük olan kayıtları siler.
* **Örnek:** DELETE FROM ogrenci WHERE adi like '\_a%'; sorgusunun işlevini yazınız.
* İsminin ikinci harfi a olan kayıtları siler.
* **Örnek:** 1 numaralı bölümdeki öğrencilerin tüm not bilgilerini notlar tablosundan silen SQL komutunu yazınız.
* DELETE FROM notlar WHERE ogr\_no IN(SELECT ogr\_no FROM ogrenci WHERE bolkod= 1)
* UPDATE ogrenci SET donem=2 WHERE donem=1; sorgusunun işlevini yazınız.
* 1.dönemdeki tüm öğrencileri 2.döneme atadı.
* **Örnek:** UPDATE ogrenci SET sinif=3 WHERE ort>49 AND sinif=2; sorgusunun işlevini yazınız.
* Sınıf ortalaması 49 ‘dan büyük olan ve 2.sınıf öğrencilerini 3.sınıfa atadı.
* **Örnek:** UPDATE ogrenci SET ort=50 WHERE ort<50 AND ort>45; sorgusunun işlevini yazınız.
* Notu 45 ile 50 arasında olan öğrencilerin notlarını 50 yapar.
* **Örnek:** Optik kodu 222 olan dersin final notlarını %10 artıran güncelleştirmeyi yapınız.
* update notlar set final=final\*1.1 where op\_kod=222
* UPDATE ogrenci SET bolumu=531;
* sorgusunun işlevini yazınız.
* Bolumu kolonunu 531 olarak değiştirir.
* Adı Feyza olan öğrencinin aldığı rehberlik dersinin vizesine 10 puan ekleyen güncelleştirmeyi yapınız.
* update notlar set vize=vize+10 where no in(select no from ogrenci where adi='feyza')
* Öğrenci tablosunun içerisine öğrencinin yaş bilgilerini de eklemek istiyorsak;
* ALTER TABLE ogrenci ADD yas INT NOT NULL;
* UPDATE ogrenci SET yas=10 Where Adi=’MURAT’
* INTO ogrenci(yas) VALUES(10);
* **Örnek:** ALTER TABLE ogrenci ADD baba\_adi CHAR(15); Sorgusunun işlevini yazınız.
* Öğrenci tablosuna baba\_adi diye char tipinde bir kolon ekler.
* **:** ALTER TABLE ogrenci ALTER COLUMN adi CHAR(15); Sorgusunun işlevini yazınız.
* Öğrenci tablosundaki adı alanının veri uzunluğunu 15 olarak değiştirir.
* **Örnek:** ALTER TABLE ogrenci ALTER COLUMN adi INT; Sorgusunun işlevini yazınız.
* Öğrenci tablosundaki adi alanının veri tipini int olarak değiştirir.
* **Örnek:** ALTER TABLE ogrenci ALTER COLUMN notu FLOAT; Sorgusunun işlevini yazınız.
* Öğrenci tablosundaki notu alanının veri tipini float olarak değiştirir.
* **Örnek:** ALTER TABLE ogrenci ALTER COLUMN cins BIT; Sorgusunun işlevini yazınız.
* Öğrenci tablosundaki cins kolonunun veri tipini bit olarak değiştirir.
* **Örnek:**ALTER TABLE ogrenci DROP COLUMN yas; sorgusunun işlevini yazınız.
* Öğrenci tablosundaki yaş kolonunu siler.
* **Örnek:** ALTER TABLE personel DROP COLUMN cinsiyet; sorgusunun işlevini yazınız.
* Personel tablosundaki cinsiyet kolonunu siler.
* EXEC SP\_RENAME 'ogrenci.adi','og\_adi'; sorgusunun işlevini yazınız.
* Öğrenci tablosundaki adi alanını ogr\_adi olarak değiştirir. EXEC SP\_RENAME 'personel.cins', 'cinsiyet'; sorgusunun işlevini yazınız.
* Personel tablosundaki cins kolonunun adını cinsiyet olarak değiştirir.
* SELECT cinsiyet, AVG(yas) FROM ogrenci GROUP BY cinsiyet; sorgusunun işlevini yazınız.
* Erkeklerin ayrı, bayanların ayrı yaş ortalamasını alarak ekrana yazar.
* SELECT ders\_adi, MAX(final) FROM notlar GROUP BY ders\_adi; sorgusunun işlevini yazınız.
* Her ders için o dersten alınan en yüksek notu listeler.
* Dersleri ve bu derslerden alınan vize notlarının ortalaması listeleyen sorguyu yazınız.
* SELECT ders\_adi, AVG(vize) FROM notlar GROUP BY ders\_adi;
* SELECT bol\_no, cinsiyet, AVG(notu) FROM notlar GROUP BY cinsyet, bol\_no;
* Her bölümde vize notu en yüksek olan kişinin notunu veren sorgu ve çıktısı aşağıdaki gibi olur.
* SELECT bolkod, MAX(vize) FROM notlar GROUP BY bol\_no;
* Her bölümdeki bayanların notları ortalamasını bulan sorguyu ve çıktıyı yazınız.
* Select bolkod,AVG(vize),AVG(final), from notlar where ogr\_no in(select ogr\_no from ogrenci where cinsiyet=‘kadın’) group by bolkod
* SELECT barkod, urunadi FROM urunler WHERE barkod NOT IN (SELECT barkod FROM urunsatis WHERE DATEDIFF(dd, tarih, GETDATE())<=30) sorgusunun işlevini yazınız.
* 1 ay içerisinde satılmayan ürünleri listeler.
* Satılan ürünlerin satılması üzerinden geçen yıl, ay ve günü hesaplayınız.
* Select \*,
* DATEDIFF(YEAR,tarih,GETDATE()) as [Kaç Sene Geçti],
* DATEDIFF(MONTH,tarih,GETDATE()) as [Kaç Ay Geçti],
* DATEDIFF(DAY,tarih,GETDATE()) as [Kaç Gün Geçti]
* from urun\_satis
* Ortalama finalleri 60’a eşit ya da büyük olan öğrenci numaralarını ve final ortalamalarını öğreci numaraları küçükten büyüğe olacak şekilde listeleyin.
* **select ogr\_no,AVG(final) from notlar**
* **group by ogr\_no having AVG(final)>=60 order by ogr\_no**
* Toplam Satıs\_adedi 1000’in üzerinde olan ürünlerin barkodlarını ve Satıs toplamlarını listeleyin.
* SELECT barkod, SUM(satis\_adedi)FROM urun\_satis GROUP BY barkod HAVING SUM(satis\_adedi)>1000)
* Toplam Satıs\_adedi 1000’in üzerinde olan ürünlerin adlarını listeleyin.
* SELECT urun\_adi FROM urunler WHERE barkod IN (SELECT barkod FROM urun\_satis GROUP BY barkod HAVING SUM(satis\_adedi)>1000)
* Örneğin tartışma bölümünde sorulan soruları ve bu sorulara verilen cevapları listelemek için;
* SELECT tartisma\_sorular.soru,tartisma\_cevaplar.cevap FROM tartisma\_sorular INNER JOIN tartisma\_cevaplar ON tartisma\_sorular.id=tartisma\_cevaplar.soruno;
* Tablolara göre sipariş veren müşterilerin isim, sipariş verdikleri tarih ve sipariş miktarı bilgisini listele.
* SELECT Musteriler.Ad, siparişler.Miktar, siparişler.Tarih
* FROM Musteriler
* INNER JOIN Siparisler
* ON Musteriler.ID = Siparisler. musteriID;
* Aynı şehre ait satıcı ve müşteri için satış elemanı adı, müşteri adı ve şehirlerini içeren bir liste hazırlamak için bir SQL ifadesi yazın.
* SELECT satici.adi AS "Satıcı",
* musteri.musteri\_adi, musteri.sehir
* FROM satici inner join musteri on
* satici.s\_id=musteri.satici\_id
* WHERE musteri.sehir=satici.sehir
* 23-25.2 arası sipariş tutarları olan siparişler için sipariş no, satın alma tutarı, müşteri adı ve şehirlerini içeren bir liste yapmak için SQL deyimi yazınız.
* SELECT siparişler.satınalma\_orani,
* musteri.musteri\_adi,musteri.sehir
* FROM musteri
* inner join siparisler on musteri.m\_id=siparisler.musteri\_id
* where satınalma\_orani BETWEEN 23 AND 25.2;
* **Örnek:** SELECT \*FROM ogrenci, notlar; sorgusunun işlevini yazınız.
* **Örnek:** Optik kodu 121 olan dersi alan öğrencilerin adlarını listeleyen sorguyu yazınız.
* SELECT ogrenci.adi FROM ogrenci, notlar WHERE ogrenci.no= notlar.og\_no AND notlar.op\_kod=121
* Ya da
* Select ogrenci.adi from ogrenci inner join notlar on
* ogrenci.no= notlar.og\_no where notlar.op\_kod=121
* **Örnek:** Dersleri ve ortalamaları veren sorguyu yazınız.
* SELECT dersler.d\_adi, AVG(notlar.ort) FROM dersler, notlar WHERE dersler. Op\_kod=notlar.op\_kod GROUP BY d\_adi
* **Örnek:** Öğrencilerin maksimum ortalamalarını veren sorguyu yazınız.
* SELECT ogrenci.adi, MAX(notlar.ort) FROM ogrenci, notlar, WHERE ogrenci.no=notlar.no GROUP BY adi
* Aldığı derslerin ortalaması 50’nin üzerinde olmasına rağmen finalde 50’nin altında not alan ve kalan öğrencilerin tüm bilgilerini veren sorguyu yazınız.
* SELECT \* FROM ogrenci, notlar, dersler WHERE notlar.og\_no=ogrenci.no AND dersler.op\_kod=notlar.op\_kod AND notlar.final<50 AND (notlar.vize+notlar.final)/2>50
* **Örnek:** Her öğrencinin finalde aldığı en yüksek notu bulan sorguyu yazınız.
* SELECT ogrenci.adi, MAX(notlar.final) FROM ogrenci, notlar WHERE notlar.og\_no=ogrenci.no GROUP BY ogrenci.adi;
* **Örnek:** Bölüm kodu531 olan öğrencinin memleket adlarını bulan sorguyu yazınız.
* select ogrenci.adi, ogrenci.soyadi,memleket.m\_adi from ogrenci,memleket where memleket.m\_no=ogrenci.tr\_kod and
* Finalden en yüksek notu alan öğrencinin adını, hangi dersten aldığını ve final notunu bulan sorguyu yazınız.
* select öğrenci.adi,dersler.ders\_adi,notlar.final from ogrenci,notlar,dersler where ogrenci.no=notlar.no and dersler.op\_kod=notlar.op\_kod and final=(select MAX(final) from notlar)
* 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);
* 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);
* Egitselyazilim veri tabanındaki kullanıcıları adı soyadı ve sordukları soruları görüntüleyen bir liste oluşturun. Soru sormayan öğrencilerin ad ve soyadları da bu listede olsun;
* SELECT kullanici\_bilgileri.ad, kullanici\_bilgileri.soyad, tartisma\_sorular.soru FROM kullanici\_bilgileri LEFT OUTER JOIN tartisma\_sorular ON kullanici\_bilgileri.ogrno=tartisma\_sorular.soran\_kisi;
* Egitselyazilim veri tabanındaki kullanıcıları adı soyadı ve sordukları soruları görüntüleyen bir liste oluşturun. Soru sormayan öğrencilerin ad ve soyadları da bu listede olsun;
* SELECT kullanici\_bilgileri.ad, kullanici\_bilgileri.soyad, tartisma\_sorular.soru FROM tartisma\_sorular RIGHT OUTER JOIN kullanici\_bilgileri ON kullanici\_bilgileri.ogrno=tartisma\_sorular.soran\_kisi;
* - her müşterinin yaptığı harcamalar toplamı SELECT mno, sum(fiyat\*adet) as Tutar FROM satış GROUP BY mno
* - stoka adına ve son kullanma tarihine göre gruplandırarak listele SELECT stokadi, sonkultar FROM stok GROUP BY stokadi, sonlultar ORDER BY sonkultar
* -- stok tablosundaki sayısı 1 den çok olan malların grup halinde stok numarası ve kalan adet toplamı SELECT sno, sum(adet) as Adedi FROM stok GROUP BY sno Having count(\*) >1
* -- stok tablosundaki fiyatı 12 liradan fazla olan malların grup halinde stok numarasına göre gruplu bir şekilde fiyat toplamını listele SELECT sno, sum(fiyat) FROM stok GROUP BY sno HAVING fiyat>12
* - öğrenci tablosunda her bölümün en yüksek ortalamalar SELECT bolum, Max( vize\*0.2+final\*0.8) FROM ogrenci GROUP BY bolum
* -- öğrenci tablosunda her bölümün, finali 50 den büyük olanlar içinden en yüksek ortalamalar SELECT ulke, sehir FROM ogrenci\_bilgi GROUP BY ulke, sehir
* -- öğrenci tablosunda her bölümün, finali 50 den büyük olanlar içinden en yüksek ortalamalar SELECT bolum, Max( vize\*0.2+final\*0.8) FROM ogrenci WHERE final>50 GROUP BY bolum