

MAKÜ BBBF BUCAK BİLGİSAYAR VE BİLİŞİM FAKÜLTESİ BUCAK BİLGİSAYAR VE BİLİŞİM FAKÜLTESİ SINAV KAĞIDI		Ders Kodu ve Adı		22206 – Veri Tabanı Yönetim Sistemleri					
		Sınav Türü		Vize		Sınıfı	2		
		Tarihi	18.03.2025	Saati	15.00	Süresi	60 Dakika		
		CEVAP ANAHTARI							
Bölüm	Yazılım Müh.								
Değerlendirme									
Soru No	1	2	3	4	5	6	7	8	Sınav Notu
Soru Puanı	15	15	10	15	10	15	10	10	
Öğrenci Puanı									
SORULAR									
1) Aşağıdaki ifadeleri bir veya en fazla iki cümleyle açıklayınız. a) Birincil anahtar b) LEFT JOIN c) ALL/ANY 1. Soru hariç tüm soruları, sınav kâğıdının alt kısmında verilen tablolara göre cevaplayın. 2) 70'in üzerindeki tüm puanlara ait öğrenci numarası, ders adı ve puan bilgisini listeleyen ilişkisel cebir gösterimini yazınız. 3) En az bir sınavdan 60 puan almış öğrencilerin adı ve soyadını getiren sorguyu yazınız. 4) Tüm derslerin kodu, adı ve öğrenciler tarafından alınan ortalama puanını, ortalama puanına göre azalan sırada getiren SQL sorgusunu yazınız. 5) Hiç ders almamış öğrencileri listeleyen SQL sorgusunu EXISTS yapısı kullanarak yazınız. 6) Her öğrencinin derslerden aldığı puanların ortalamasına göre (her ders eşit ağırlıkta) ortalaması 70 puanın altında kalan öğrencileri ve ortalamalarını listeleyiniz. 7) Tüm notları 5 puan artıran ancak 100'ü geçmesini engelleyen bir güncelleme sorgusu yazınız. 8) Aşağıdaki sorguyu açıklayınız ve tabloları dikkate alarak sorgu sonucunu yazınız									
<pre>SELECT o.ad, o.soyad, d.dkodu, d.dadi FROM ogrenci o INNER JOIN ders_kayit dk ON o.ogr_no = dk.ogr_no INNER JOIN ders d ON dk.dkodu = d.dkodu WHERE dk.dkodu IN (SELECT dkodu FROM ders_kayit GROUP BY dkodu HAVING COUNT(DISTINCT ogr_no) = 1);</pre>									
Tablo:ogrenci			Tablo:ders			Tablo:ders_kayit			
ogr_no	ad	soyad	dkodu	dadi	ogr_no	dkodu	puan		
1001	Ahmet	Yılmaz	YAZ101	Veritabanı	1001	YAZ101	55		
1002	Ayşe	Demir	YAZ102	Algoritma	1002	YAZ101	70		
1003	Mehmet	Kaya	YAZ201	Nesne	1002	YAZ102	90		
1004	Zeynep	Çelik	YAZ202	Web	1003	YAZ201	60		
1005	Ali	Şahin	YAZ301	Cebir	1003	YAZ202	75		
1006	Sevgi	Soysal			1003	YAZ301	90		
					1004	YAZ301	50		
					1005	YAZ101	95		

CEVAP ANAHTARI

1) a) Birincil Anahtar (Primary Key): Tablodaki her bir satırı diğer tüm satırlardan benzersiz yapan nitelik(ler)dir. Günümüzde otomatik artan sıralı integer tipinde tanımlamaları yapılmaktadır.

b) LEFT JOIN: Sol tablodaki tüm kayıtları eksiksiz getirir, sağ tablo ile eşleşmeyen kayıtlar için NULL değer gösterir.

c) ALL karşılaştırmadaki tüm sonuçların doğru olması gereken **ANY** ise karşılaştırmalarından en az bir doğru sonuç döndürmesi gereken durumlarda kullanılır.

2)

$$\pi_{\text{ogr_no}, \text{dadi}, \text{puan}} (\sigma_{\text{puan} > 70} (\text{ders_kayit} \bowtie \text{ders}))$$

3) Birçok doğru cevabı bulunabilir. Bazıları şöyledir. Diğer sorularda uygun bir çözüm sunulacaktır.

<pre>SELECT O.ad, O.soyad FROM ogrenci AS O WHERE EXISTS (SELECT * FROM ders_kayit AS DK WHERE DK.ogr_no = O.ogr_no AND DK.puan >= 60);</pre>	<pre>SELECT ad, soyad FROM ogrenci WHERE ogr_no IN (SELECT ogr_no FROM ders_kayit WHERE puan >= 60);</pre>	<pre>SELECT DISTINCT o.ad, o.soyad FROM ogrenci o JOIN ders_kayit dk ON o.ogr_no = dk.ogr_no WHERE dk.puan >= 60;</pre>	<table><tr><th>ad</th><th>soyad</th></tr><tr><td>Ayşe</td><td>Demir</td></tr><tr><td>Mehmet</td><td>Kaya</td></tr><tr><td>Ali</td><td>Şahin</td></tr></table>	ad	soyad	Ayşe	Demir	Mehmet	Kaya	Ali	Şahin
ad	soyad										
Ayşe	Demir										
Mehmet	Kaya										
Ali	Şahin										

4)

SELECT D.dkodu, D.dadi, AVG(DK.puan) AS ort_puan FROM ders AS D JOIN ders_kayit AS DK ON D.dkodu = DK.dkodu GROUP BY D.dkodu ORDER BY ort_puan DESC;	<table><tr><th>dkodu</th><th>dadi</th><th>ort_puan</th><th>▼ 1</th></tr><tr><td>YAZ102</td><td>Algoritma</td><td>90.0000</td><td></td></tr><tr><td>YAZ202</td><td>Web</td><td>75.0000</td><td></td></tr><tr><td>YAZ101</td><td>Veritabanı</td><td>73.3333</td><td></td></tr><tr><td>YAZ301</td><td>Cebir</td><td>70.0000</td><td></td></tr><tr><td>YAZ201</td><td>Nesne</td><td>60.0000</td><td></td></tr></table>	dkodu	dadi	ort_puan	▼ 1	YAZ102	Algoritma	90.0000		YAZ202	Web	75.0000		YAZ101	Veritabanı	73.3333		YAZ301	Cebir	70.0000		YAZ201	Nesne	60.0000	
dkodu	dadi	ort_puan	▼ 1																						
YAZ102	Algoritma	90.0000																							
YAZ202	Web	75.0000																							
YAZ101	Veritabanı	73.3333																							
YAZ301	Cebir	70.0000																							
YAZ201	Nesne	60.0000																							

5)

SELECT * FROM ogrenci AS O WHERE NOT EXISTS (SELECT * FROM ders_kayit AS DK WHERE DK.ogr_no = O.ogr_no);	<table><tr><th>ogr_no</th><th>ad</th><th>soyad</th></tr><tr><td>1006</td><td>Sevgi</td><td>Soysal</td></tr></table>	ogr_no	ad	soyad	1006	Sevgi	Soysal
ogr_no	ad	soyad					
1006	Sevgi	Soysal					

6)

SELECT O.ad, O.soyad, AVG(DK.puan) AS ortalama FROM ogrenci AS O JOIN ders_kayit AS DK ON O.ogr_no = DK.ogr_no GROUP BY O.ogr_no HAVING AVG(DK.puan)<70;	<table><tr><th>ad</th><th>soyad</th><th>ortalama</th></tr><tr><td>Ahmet</td><td>Yılmaz</td><td>55.0000</td></tr><tr><td>Zeynep</td><td>Çelik</td><td>50.0000</td></tr></table>	ad	soyad	ortalama	Ahmet	Yılmaz	55.0000	Zeynep	Çelik	50.0000
ad	soyad	ortalama								
Ahmet	Yılmaz	55.0000								
Zeynep	Çelik	50.0000								

7)

UPDATE ders_kayit SET puan = puan + 5 WHERE puan <= 95; UPDATE ders_kayit SET puan = 100 WHERE puan > 95;	Farklı Yöntemler UPDATE ders_kayit SET puan = IF(puan + 5 > 100, 100, puan + 5); UPDATE ders_kayit SET puan = CASE WHEN puan + 5 > 100 THEN 100 ELSE puan + 5 END;
--	--

8 –

Yalnızca bir öğrenci tarafından alınmış derslerin kodu ve adıyla ile bu dersleri alan öğrencilerin adı soyadı,	ad	soyad	dkodu	dadi
	Ayşe	Demir	YAZ102	Algoritma
	Mehmet	Kaya	YAZ201	Nesne
	Mehmet	Kaya	YAZ202	Web

MYSQL veri tabanı

```
CREATE TABLE `ders` (  
  `dkodu` varchar(10) NOT NULL,  
  `dadi` varchar(50) DEFAULT NULL,  
  `kredi` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

--

-- Tablo döküm verisi `ders`

--

```
INSERT INTO `ders` (`dkodu`, `dadi`, `kredi`) VALUES
```

```
('YAZ101', 'Veritabanı', 4),
```

```
('YAZ102', 'Algoritma', 3),
```

```
('YAZ201', 'Nesne', 4),
```

```
('YAZ202', 'Web', 3),
```

```
('YAZ301', 'Cebir', 3);
```

--

-- Tablo için tablo yapısı `ders\_kayit`

--

```
CREATE TABLE `ders_kayit` (
```

```
`ogr_no` int(11) NOT NULL,  
`dkodu` varchar(10) NOT NULL,  
`puan` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
--
```

```
-- Tablo döküm verisi `ders_kayit`
```

```
--
```

```
INSERT INTO `ders_kayit` (`ogr_no`, `dkodu`, `puan`) VALUES
```

```
(1001, 'YAZ101', 55),
```

```
(1002, 'YAZ101', 70),
```

```
(1002, 'YAZ102', 90),
```

```
(1003, 'YAZ201', 60),
```

```
(1003, 'YAZ202', 75),
```

```
(1003, 'YAZ301', 90),
```

```
(1004, 'YAZ301', 50),
```

```
(1005, 'YAZ101', 95);
```

```
-- -----
```

```
--
```

```
-- Tablo için tablo yapısı `ogrenci`
```

```
--
```

```
CREATE TABLE `ogrenci` (  
`ogr_no` int(11) NOT NULL,
```

```
`ad` varchar(50) DEFAULT NULL,
`soyad` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--

-- Tablo döküm verisi `ogrenci`

--

INSERT INTO `ogrenci` (`ogr_no`, `ad`, `soyad`) VALUES
(1001, 'Ahmet', 'Yılmaz'),
(1002, 'Ayşe', 'Demir'),
(1003, 'Mehmet', 'Kaya'),
(1004, 'Zeynep', 'Çelik'),
(1005, 'Ali', 'Şahin'),
(1006, 'Sevgi', 'Soysal');

--

-- Dökümü yapılmış tablolar için indeksler

--

-- Tablo için indeksler `ders`

--

ALTER TABLE `ders`

ADD PRIMARY KEY (`dkodu`);

--
```

```
-- Tablo için indeksler `ders_kayit`
```

```
--
```

```
ALTER TABLE `ders_kayit`
```

```
ADD PRIMARY KEY (`ogr_no`,`dkodu`),
```

```
ADD KEY `dkodu` (`dkodu`);
```

```
--
```

```
-- Tablo için indeksler `ogrenci`
```

```
--
```

```
ALTER TABLE `ogrenci`
```

```
ADD PRIMARY KEY (`ogr_no`);
```

```
--
```

```
-- Dökümü yapılmış tablolar için kısıtlamalar
```

```
--
```

```
--
```

```
-- Tablo kısıtlamaları `ders_kayit`
```

```
--
```

```
ALTER TABLE `ders_kayit`
```

```
ADD CONSTRAINT `ders_kayit_ibfk_1` FOREIGN KEY (`ogr_no`) REFERENCES `ogrenci`  
(`ogr_no`) ON DELETE CASCADE,
```

```
ADD CONSTRAINT `ders_kayit_ibfk_2` FOREIGN KEY (`dkodu`) REFERENCES `ders` (`dkodu`)  
ON DELETE CASCADE;
```

```
COMMIT;
```