

BIMU3064-Veri Tabanı Yönetim Sistemleri

**İBRAHİM BAŞAR YARGICI**

**1306191467**

Ödev:1

Veritabanı = PostgreSQL

(28.10.2021)

**İÇERİK**

1. **Kapak ve Giriş ……………………………………………..………………………………………………………..1**
2. **Veri Tabanı ve Veri Tabanı Şeması Hazırlığı ..……………….………………………………………..4**
3. [Sorgular, Relational Algebra gösterimleri ve çıktılar](#_TOC_250008) 10
   1. [Sorgu 1](#_TOC_250006) 11
   2. [Sorgu 2](#_TOC_250006) 14
   3. [Sorgu 3](#_TOC_250006) 14
      1. Where Koşulu Ile 14
      2. Natural Join Ile 15
      3. Where Koşulu, IN Terimi Ile 16
      4. Some Ile 17
      5. Exists Ile 18
   4. [Sorgu 4](#_TOC_250006) 18
      1. Except Ile 19
      2. Not In Ile 20
      3. != All Ile 21
      4. Not Exists Ile 22
   5. [Sorgu 5](#_TOC_250006) 23
   6. [Sorgu 6](#_TOC_250006) 24
   7. [Sorgu 7](#_TOC_250006) 25
   8. [Sorgu 8](#_TOC_250006) 27
   9. [Sorgu 9](#_TOC_250006) 28

2.10 Sorgu 10 29

**ÖDEV 1**

**VERİTABANI ŞEMASI**

sarki(sarkino, ad, tur, sure, bestecino, yazarno)

sarkici(sarkicino, ad, tur, dogumTarih, dogumYer)

album(albumno, ad, yil, fiyat, sarkicino, stokAdet)

albumdekiSarki(albumno, sarkino, sira)

besteci(bestecino, ad, tur)

sozyazari(yazarno, ad)

**SORGULAR** (Soruların hepsi SQL ve İlişkisel Cebir’le cevaplanacak. Dikaat 3. ve 4. sorularda sadece SQL için 5 ve 4 farklı çözüm istenmektedir.)

Önce yukarıdaki tabloları CREATE TABLE ile oluşturunuz. Sonra tablolara rastgele kayıtlar INSERT ediniz. Sonra sorgularınızı yazıp test ediniz.

1. Türü ‘arabesk’ ve süresi 3 dakikadan uzun olan şarkıların adlarını tekrarsız olarak ve alfanümerik sırada artan sırada listeleyiniz. [Not: RA sorgusunda sıralamayı göz önüne almayınız. RA’da sıralama işlemi yoktur]
2. Doğum yeri ‘Ankara’ olan ya da albüme sahip olan şarkıcıların kayıtlarını listeleyiniz. UNION kullanırken sarkicino kullanılarak UNION yapınız. ŞArkıcı kayıtlarını getirmek için UNION işleminin sonucunu sarkici tablosuyla join (kartezyen çarpım ve kayıt seçme işlemi) ediniz
3. Şarkıcılar ve albümlerini *şarkıcının adı, albümünün adı, album yılı* şeklinde listeleyen sorguyu WHERE içinde sarkici.sarkicino=album.sarkicino koşuluyla, FROM içinde NATURAL JOIN cümleciğiyle, WHERE içinde IN ile, =SOME ile ve EXISTS kullanarak 5 farklı şekilde yazınız.
4. Hiç albümü olmayan şarkıcıların kayıtlarını küme farkı (MINUS/EXCEPT), NOT IN, != ALL ve NOT EXISTS kullanarak 4 farklı şekilde yazınız.
5. [GROUP BY] Albümlerin numaraları, adı, şarkıcısının adı, fiyatı, içindeki şarkıların sayısı ve süre uzunlukları toplamını listeleyiniz.
6. [GROUP BY, HAVING] ‘arabesk’ türündeki bestecilerden 5 adetten çok şarkı besteleyenlerin besteci adı ve kaç şarkı bestelediklerini listeleyiniz. [Not: İlk adımda ‘arabesk’ türündeki bestecilerin kaç şarkı besteledikleri besteci ve sarki tabloları join edilerek bulunabilir. Besteci tablosunun burada kullanılmasının sebebi *besteci.tur=’arabesk’* koşulunu sağlayan bestecileri bulabilmektir. Sonraki adımda bestecilerin adları besteci tablosundan bulunur]
7. En fazla albüme sahip şarkıcıların kayıtlarını listeleyiniz. [Not:Önce her şarkıcının album sayısı GROUP BY ile bulunur, sonra en yüksek album sayısı bulunur. Sonra en yüksek album sayısına sahip şarkıcıların kayıtları bulunur. Ön adımdaki sorguları WITH ile yazınız]
8. ‘arabesk’ türündeki tüm şarkıların yazarlığını yapmış yazarların kayıtlarını listeleyiniz. [Not: İlişkisel cebirdeki bölme (÷) işlemi kullanılacak ve ilk adımda yazarların numaraları bulunduktan sonra ikinci adımda yazar kayıtları bulunacak. Bu sorguyu dersteki *tüm dersleri alan öğrenciler* sorgusuna benzeterek çözebilirsiniz]
9. En çok şarkı sözü yazmış olan yazarların yazarno’larını aggregate fonksiyon veya group by kullanmadan sınıfta (sunumdaki örnekte) çözdüğümüz gibi çözünüz.
10. En az iki şarkı bestelemiş olan bestecilerin bestecino’larını aynı tabloyu kendisi ile kartezyen çarpım yaparak sınıfta (sunumdaki örnekte) olduğu gibi çözünüz.

**Veri Tabanı ve Veri Tabanı Şeması Hazırlığı**

**(bu sayfa bilerek boş bırakılmıştır)**

|  |
| --- |
| CREATE DATABASE "Homework\_28102021";  -- CREATE TABLE  DROP TABLE IF EXISTS besteci CASCADE;  DROP TABLE IF EXISTS sozyazari CASCADE;  DROP TABLE IF EXISTS sarkici CASCADE;  DROP TABLE IF EXISTS album CASCADE;  DROP TABLE IF EXISTS sarki CASCADE;  DROP TABLE IF EXISTS albumdekiSarki CASCADE;  -- besteci(bestecino, ad, tur)  CREATE TABLE besteci(  bestecino INTEGER NOT NULL PRIMARY KEY,  ad VARCHAR(50),  tur VARCHAR(50)  );  -- sozyazari(yazarno, ad)  CREATE TABLE sozyazari(  yazarno INTEGER NOT NULL PRIMARY KEY,  ad VARCHAR(50)  );  -- sarkici(sarkicino, ad, tur, dogumTarih, dogumYer)  CREATE TABLE sarkici(  sarkicino INTEGER NOT NULL PRIMARY KEY,  ad VARCHAR(50),  tur VARCHAR(50),  dogumTarih date,  dogumYer VARCHAR(50)  );  -- album(albumno, ad, yil, fiyat, stokAdet, sarkicino)  CREATE TABLE album(  albumno INTEGER NOT NULL PRIMARY KEY,  ad VARCHAR(50),  yil VARCHAR(50),  fiyat INTEGER,  stokAdet INTEGER,  sarkicino INTEGER NOT NULL,  FOREIGN KEY(sarkicino) REFERENCES sarkici(sarkicino)  );  -- sarki(sarkino, ad, tur, sure, bestecino, yazarno)  CREATE TABLE sarki(  sarkino INTEGER NOT NULL PRIMARY KEY,  ad VARCHAR(50),  tur VARCHAR(50),  sure INTEGER,  bestecino INTEGER NOT NULL,  yazarno INTEGER NOT NULL,  FOREIGN KEY(bestecino) REFERENCES besteci(bestecino),  FOREIGN KEY(yazarno) REFERENCES sozyazari(yazarno)  );  -- albumdekiSarki(sira, albumno, sarkino)  CREATE TABLE albumdekiSarki(  sira INTEGER,  albumno INTEGER NOT NULL,  sarkino INTEGER NOT NULL,  PRIMARY KEY(albumno,sarkino),  FOREIGN KEY(albumno) REFERENCES album(albumno),  FOREIGN KEY(sarkino) REFERENCES sarki(sarkino)  );  -- TARKAN  -- sarkici(sarkicino, ad, tur, dogumTarih, dogumYer)  INSERT INTO sarkici VALUES(1,'Tarkan', 'Pop', '1972-OCT-17','Istanbul');  -- besteci(bestecino, ad, tur)  INSERT INTO besteci VALUES(1,'Ozan Çolakoğlu', 'Pop');  INSERT INTO besteci VALUES(2,'Jeff Koplan', 'Pop');  INSERT INTO besteci VALUES(3,'Nazan Öncel', 'Pop');  -- sozyazari(yazarno, ad)  INSERT INTO sozyazari VALUES(1,'Nazan Öncel');  INSERT INTO sozyazari VALUES(2,'Tarkan');  -- sarki(sarkino, ad, tur, sure, bestecino, yazarno)  INSERT INTO sarki VALUES(1,'Dudu', 'Pop', 275, 2, 1);  INSERT INTO sarki VALUES(2,'Bu Şarkılar da Olmasa', 'Pop', 266, 3, 1);  INSERT INTO sarki VALUES(3,'Sorma Kalbim', 'Pop', 275, 1, 2);  -- album(albumno, ad, yil, fiyat, stokAdet, sarkicino)  INSERT INTO album VALUES(1,'Dudu',2003, 99, 150,1);  INSERT INTO album VALUES(4,'Olurum Sana',1997, 201, 100,1);  -- albumdekiSarki(sira, albumno, sarkino)  INSERT INTO albumdekiSarki VALUES(1,1,1);  INSERT INTO albumdekiSarki VALUES(2,1,2);  INSERT INTO albumdekiSarki VALUES(3,1,3);  -- Müslüm Gürses  -- sarkici(sarkicino, ad, tur, dogumTarih, dogumYer)  INSERT INTO sarkici VALUES(2,'Müslüm Gürses', 'Arabesk', '1953-JUL-5','Ankara');  -- besteci(bestecino, ad, tur)  INSERT INTO besteci VALUES(4,'Teoman', 'Arabesk');  INSERT INTO besteci VALUES(5,'Ali Mamaraşlı', 'Arabesk');  INSERT INTO besteci VALUES(6,'Ahmet Selçuk İlkan', 'Arabesk');  -- sozyazari(yazarno, ad)  INSERT INTO sozyazari VALUES(3,'Teoman');  INSERT INTO sozyazari VALUES(4,'Mehmet Aslan');  INSERT INTO sozyazari VALUES(5,'Burhan Bayar');  INSERT INTO sozyazari VALUES(6,'Yunus Emre');  INSERT INTO sozyazari VALUES(7,'Tahir Paker');  -- sarki(sarkino, ad, tur, sure, bestecino, yazarno)  INSERT INTO sarki VALUES(4,'Paramparça', 'Arabesk', 215 , 4, 3);  INSERT INTO sarki VALUES(5,'Sen Yoksun', 'Arabesk', 207 , 5, 4);  INSERT INTO sarki VALUES(6,'Yitik Aşk', 'Arabesk', 112 , 5, 5);  INSERT INTO sarki VALUES(7,'Çay Karam', 'Arabesk', 306, 6, 5);  INSERT INTO sarki VALUES(8,'Aşk Bir Ateş', 'Arabesk', 239, 4, 6);  INSERT INTO sarki VALUES(9,'Olan Bana Oldu', 'Arabesk', 206, 4, 7);  -- album(albumno, ad, yil, fiyat, stokAdet, sarkicino)  INSERT INTO album VALUES(2,'Paramparça',2002, 569, 18,2);  -- albumdekiSarki(sira, albumno, sarkino)  INSERT INTO albumdekiSarki VALUES(4,2,4);  INSERT INTO albumdekiSarki VALUES(5,2,5);  INSERT INTO albumdekiSarki VALUES(6,2,6);  INSERT INTO albumdekiSarki VALUES(7,2,7);  INSERT INTO albumdekiSarki VALUES(8,2,8);  INSERT INTO albumdekiSarki VALUES(9,2,9);  -- Orhan Gencebay  -- sarkici(sarkicino, ad, tur, dogumTarih, dogumYer)  INSERT INTO sarkici VALUES(3,'Orhan Gencebay', 'Arabesk', '1944-AUG-4','SAMSUN');  -- besteci(bestecino, ad, tur)  INSERT INTO besteci VALUES(7,'Orhan Gencebay', 'Arabesk');  -- sozyazari(yazarno, ad)  INSERT INTO sozyazari VALUES(8,'Orhan Gencebay');  -- sarki(sarkino, ad, tur, sure, bestecino, yazarno)  INSERT INTO sarki VALUES(10,'Emaneti Alda Götür', 'Arabesk', 310, 7, 8);  INSERT INTO sarki VALUES(12,'Vatan Sağolsun', 'Arabesk', 279, 7, 8);  INSERT INTO sarki VALUES(13,'Berhudar Ol', 'Arabesk', 329, 7, 8);  INSERT INTO sarki VALUES(14,'Sevmeyi Bilmiyorlar', 'Arabesk', 225, 7, 8);  INSERT INTO sarki VALUES(15,'Sensizlik Yerle Bir Etti', 'Arabesk', 322, 7, 8);  INSERT INTO sarki VALUES(16,'Her gün Başka Başka', 'Arabesk', 190, 7, 8);  INSERT INTO sarki VALUES(17,'Yalnızlık', 'Arabesk', 389, 7, 8);  INSERT INTO sarki VALUES(18,'Sen de mi Kayboldun', 'Arabesk', 266, 7, 8);  INSERT INTO sarki VALUES(19,'Duymadın Ki', 'Arabesk', 363, 7, 8);  INSERT INTO sarki VALUES(20,'Diriliş /Etrüksler (Enstrümantal)', 'Arabesk', 394, 7, 8);  -- album(albumno, ad, yil, fiyat, stokAdet, sarkicino)  INSERT INTO album VALUES(3,'Berhudar Ol',2010, 324, 547, 3);  -- albumdekiSarki(sira, albumno, sarkino)  INSERT INTO albumdekiSarki VALUES(10,3,10);  INSERT INTO albumdekiSarki VALUES(12,3,12);  INSERT INTO albumdekiSarki VALUES(13,3,13);  INSERT INTO albumdekiSarki VALUES(14,3,14);  INSERT INTO albumdekiSarki VALUES(15,3,15);  INSERT INTO albumdekiSarki VALUES(16,3,16);  INSERT INTO albumdekiSarki VALUES(17,3,17);  INSERT INTO albumdekiSarki VALUES(18,3,18);  INSERT INTO albumdekiSarki VALUES(19,3,19);  INSERT INTO albumdekiSarki VALUES(20,3,20);  -- sarkici(sarkicino, ad, tur, dogumTarih, dogumYer)  INSERT INTO sarkici VALUES(4,'Basar Yargici', null, '2000-11-01','Istanbul');  SELECT \* FROM besteci;  SELECT \* FROM sozyazari;  SELECT \* FROM sarkici;  SELECT \* FROM album;  SELECT \* FROM sarki;  SELECT \* FROM albumdekiSarki; |

**Sorgulara Giriş**

**(sorgular soru, relational algebra gösterimi ve çıktı şeklinde sıralanmıştır)**

**(bu sayfa bilerek boş bırakılmıştır)**

Sorgu 1:

Türü ‘arabesk’ ve süresi 3 dakikadan uzun olan şarkıların adlarını tekrarsız olarak ve alfanümerik sırada artan sırada listeleyiniz. [Not: RA sorgusunda sıralamayı göz önüne almayınız. RA’da sıralama işlemi yoktur]

|  |
| --- |
| SELECT s.ad, s.sure  FROM sarki s  WHERE s.tur = 'Arabesk' AND s.sure > 3\*60  ORDER BY s.ad ASC; |

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

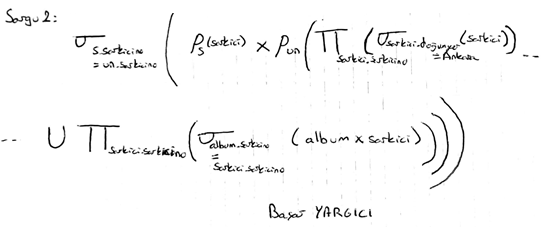
metin, ekran, siyah, skorbord içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sorgu 2:

Doğum yeri ‘Ankara’ olan ya da albüme sahip olan şarkıcıların kayıtlarını listeleyiniz. UNION kullanırken sarkicino kullanılarak UNION yapınız. ŞArkıcı kayıtlarını getirmek için UNION işleminin sonucunu sarkici tablosuyla join (kartezyen çarpım ve kayıt seçme işlemi) ediniz

|  |
| --- |
| -- 0. çözüm  SELECT \*  FROM sarkici s, (  (  SELECT sarkici.sarkicino  FROM sarkici  WHERE sarkici.dogumyer = 'Ankara'  )  UNION  (  SELECT sarkici.sarkicino  FROM album,sarkici  WHERE album.sarkicino = sarkici.sarkicino  )  ) un  WHERE s.sarkicino IN (un.sarkicino); |

 metin, ekran görüntüsü, ekran, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

|  |
| --- |
| -- Ürettiğim bir başka çözüm (dıştaki sarkici döngüsüne gerek kalmıyor):  (  SELECT sarkici.\*  FROM sarkici  WHERE sarkici.dogumyer = 'Ankara'  )  UNION  (  SELECT sarkici.\*  FROM album,sarkici  WHERE album.sarkicino = sarkici.sarkicino  ); |

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sorgu 3:

Şarkıcılar ve albümlerini *şarkıcının adı, albümünün adı, album yılı* şeklinde listeleyen sorguyu WHERE içinde sarkici.sarkicino=album.sarkicino koşuluyla, FROM içinde NATURAL JOIN cümleciğiyle, WHERE içinde IN ile, =SOME ile ve EXISTS kullanarak 5 farklı şekilde yazınız.

1. Where koşulu ile

|  |
| --- |
| -- with where  SELECT s.ad, a.ad, a.yil  FROM sarkici s, album a  WHERE s.sarkicino=a.sarkicino; |

metin içeren bir resim

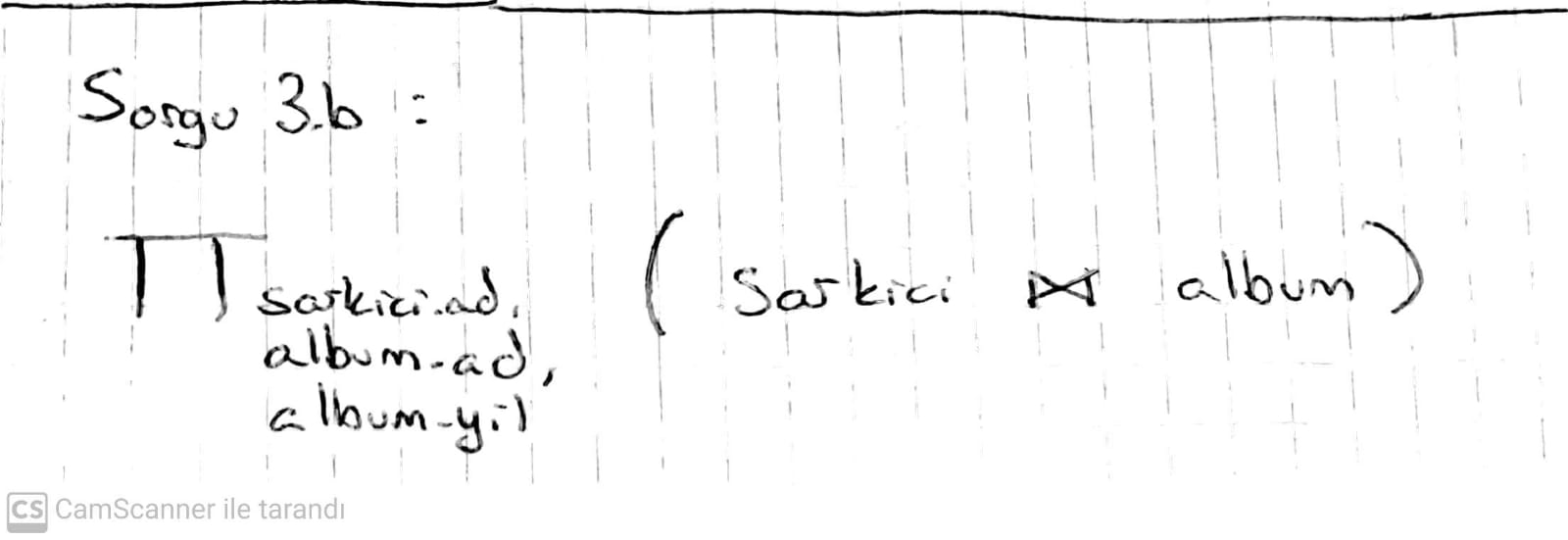
Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Natural Join cümleciği ile

|  |
| --- |
| -- with NATURAL JOIN (Sanıyorum ki iki ortak alan olduğu için bu çözüm işe yaramadı ancak aynı anlama gelen JOIN ON ile çalıştırdım.)  SELECT sarkici.ad, album.ad, album.yil  FROM (sarkici NATURAL JOIN album); |

|  |
| --- |
| SELECT sarkici.ad, album.ad, album.yil  FROM sarkici  JOIN album  ON sarkici.sarkicino = album.sarkicino; |



metin, ekran görüntüsü, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Where koşulu, IN terimi ile

|  |
| --- |
| -- with IN  SELECT s.ad, a.ad, a.yil  FROM sarkici s, album a  WHERE s.sarkicino  IN(a.sarkicino); |

metin içeren bir resim

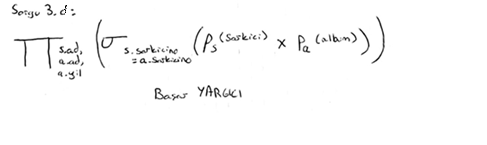
Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. Some ile

|  |
| --- |
| -- with Some/ANY  SELECT s.ad, al.ad, al.yil  FROM sarkici s, album al  WHERE s.sarkicino = some(  SELECT al.sarkicino  FROM album  WHERE s.sarkicino = al.sarkicino  ); |



metin, ekran görüntüsü, ekran içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. EXISTS ile

|  |
| --- |
| -- with EXISTS  SELECT s.ad, al.ad, al.yil  FROM sarkici s, album al  WHERE EXISTS(  SELECT al.sarkicino  FROM album  WHERE s.sarkicino = al.sarkicino  ); |

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran içeren bir resim

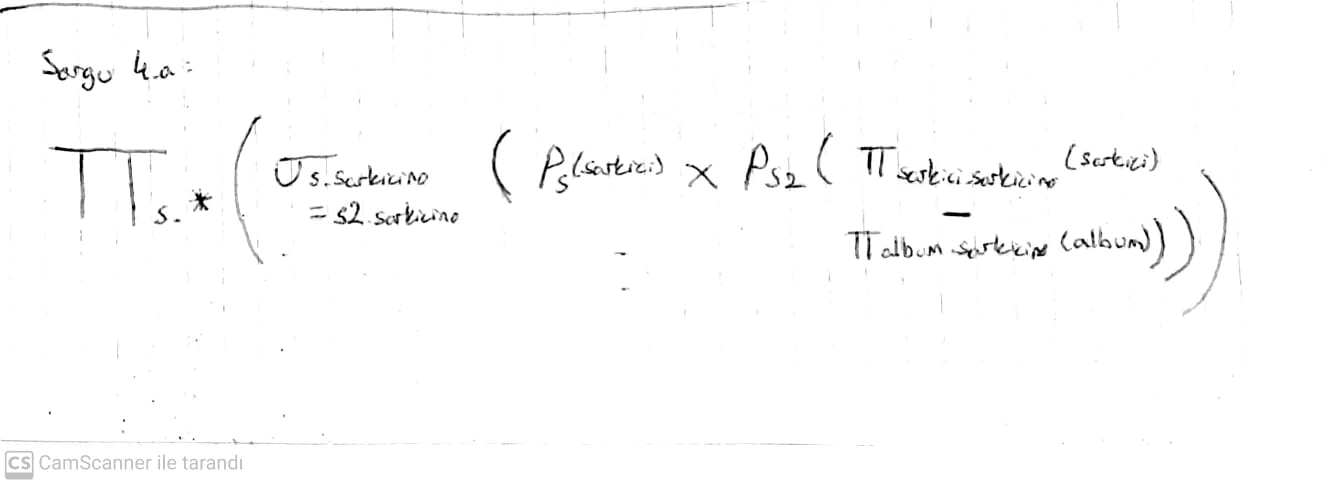
Açıklama otomatik olarak oluşturuldu

Sorgu 4:

Hiç albümü olmayan şarkıcıların kayıtlarını küme farkı (MINUS/EXCEPT), NOT IN, != ALL ve NOT EXISTS kullanarak 4 farklı şekilde yazınız.

1. EXCEPT ile

|  |
| --- |
| -- with (MINUS/EXCEPT)  SELECT s.sarkicino, s.ad, s.tur, s.dogumtarih, s.dogumyer  FROM sarkici s, (  (  SELECT sarkici.sarkicino  FROM sarkici  )  EXCEPT  (  SELECT album.sarkicino  FROM album  )  ) s2  WHERE s.sarkicino = s2.sarkicino; |

metin içeren bir resim

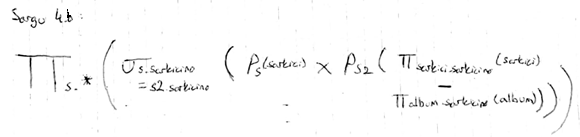
Açıklama otomatik olarak oluşturuldu(Tabloya hiç albümümün olmadığını kaydetmiştik)

metin, ekran görüntüsü, ekran, kapat içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. NOT IN ile

|  |
| --- |
| -- with NOT IN  SELECT s.\*  FROM sarkici s, (  SELECT sarkici.sarkicino  FROM sarkici  WHERE sarkici.sarkicino NOT IN  (  SELECT album.sarkicino  FROM album  )  ) s2  WHERE s.sarkicino = s2.sarkicino; |



metin içeren bir resim

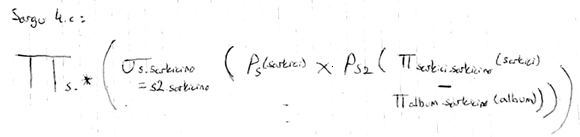
Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. != ALL ile

|  |
| --- |
| -- with != ALL  SELECT s.\*  FROM sarkici s, (  SELECT sarkici.sarkicino  FROM sarkici  WHERE sarkici.sarkicino != ALL  (  SELECT album.sarkicino  FROM album  )  ) s2  WHERE s.sarkicino = s2.sarkicino; |

metin içeren bir resim

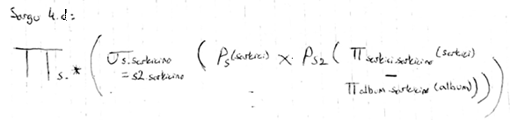
Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran, kapat içeren bir resim

Açıklama otomatik olarak oluşturuldu

1. NOT EXISTS ile

|  |
| --- |
| -- with NOT EXISTS  SELECT s.sarkicino, s.ad, s.tur, s.dogumtarih, s.dogumyer  FROM sarkici s, (  SELECT sa.sarkicino  FROM sarkici sa  WHERE NOT EXISTS  (  SELECT album.sarkicino  FROM album  WHERE sa.sarkicino = album.sarkicino  )  ) s2  WHERE s.sarkicino = s2.sarkicino; |

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

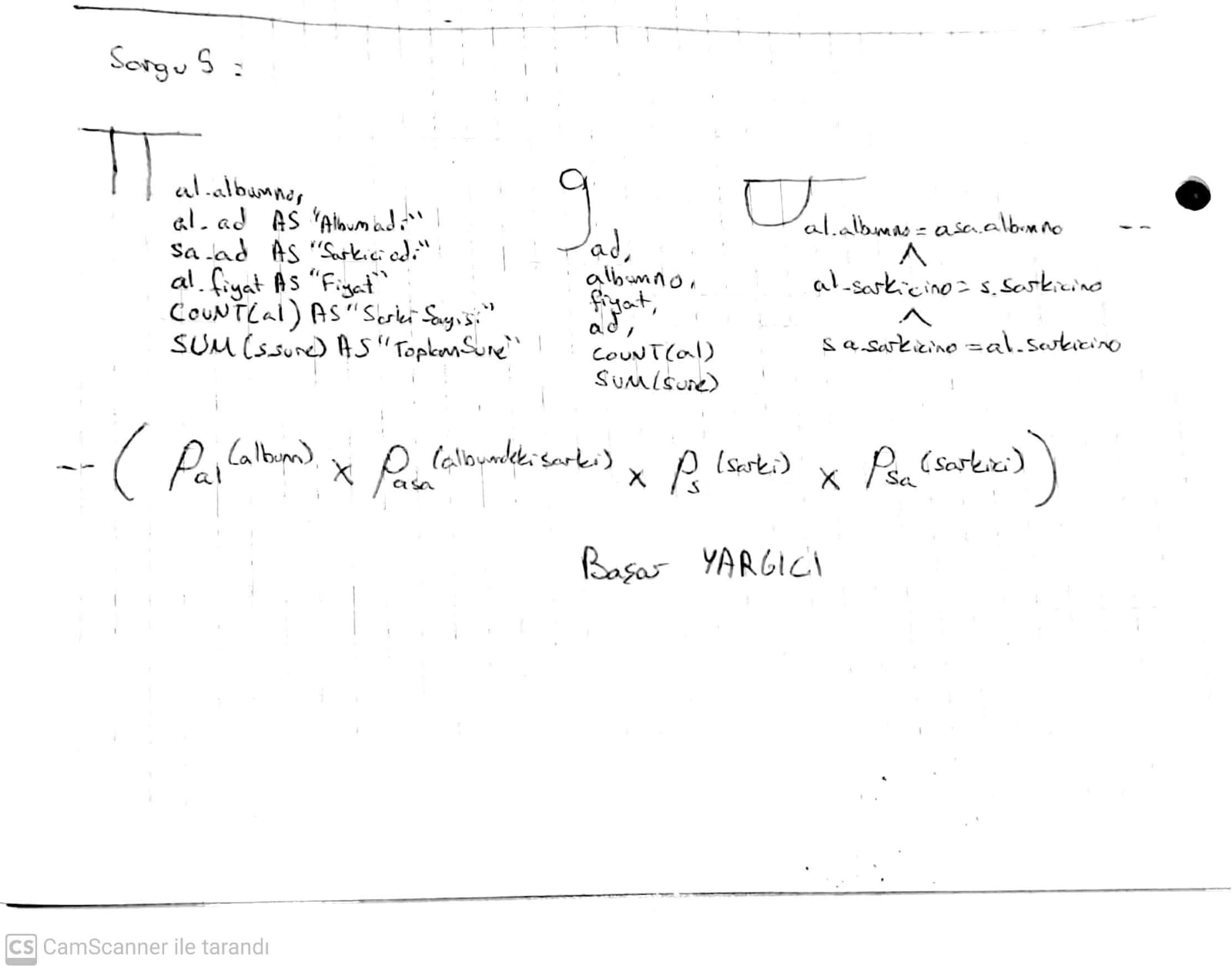
metin, ekran görüntüsü, ekran, kapat içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sorgu 5

[GROUP BY] Albümlerin numaraları, adı, şarkıcısının adı, fiyatı, içindeki şarkıların sayısı ve süre uzunlukları toplamını listeleyiniz.

|  |
| --- |
| SELECT  al.albumno, al.ad AS "Album Adi", sa.ad AS "Sarkici Adi", al.fiyat, COUNT(al) AS "Sarki sayisi", SUM(s.sure) AS "Toplam Sure"  FROM  album AS al, albumdekisarki AS asa, sarki AS s, sarkici AS sa  WHERE  al.albumno = asa.albumno AND al.sarkicino = s.sarkino AND sa.sarkicino = al.sarkicino  GROUP BY al.ad, al.albumno, al.fiyat, sa.ad; |



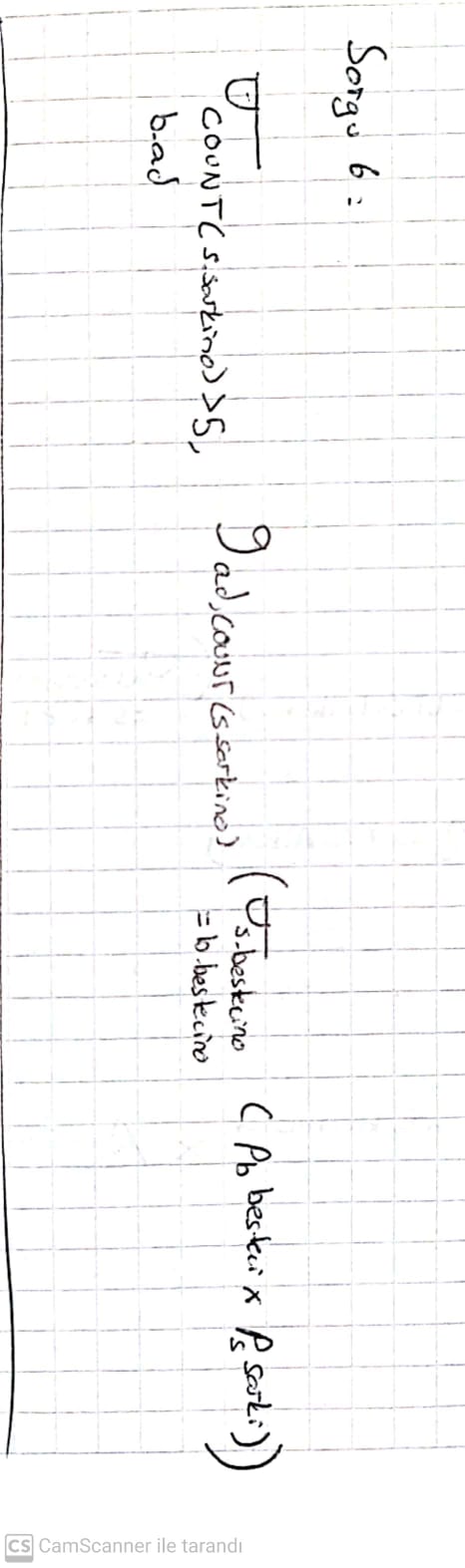
metin, ekran görüntüsü, ekran, siyah içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sorgu 6

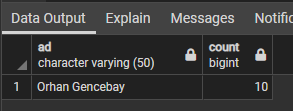
[GROUP BY, HAVING] ‘arabesk’ türündeki bestecilerden 5 adetten çok şarkı besteleyenlerin besteci adı ve kaç şarkı bestelediklerini listeleyiniz. [Not: İlk adımda ‘arabesk’ türündeki bestecilerin kaç şarkı besteledikleri besteci ve sarki tabloları join edilerek bulunabilir. Besteci tablosunun burada kullanılmasının sebebi *besteci.tur=’arabesk’* koşulunu sağlayan bestecileri bulabilmektir. Sonraki adımda bestecilerin adları besteci tablosundan bulunur]

|  |
| --- |
| SELECT b.ad, COUNT(s.sarkino)  FROM besteci b, sarki s  WHERE s.bestecino = b.bestecino  GROUP BY b.ad  HAVING COUNT(s.sarkino)>5; |



metin içeren bir resim

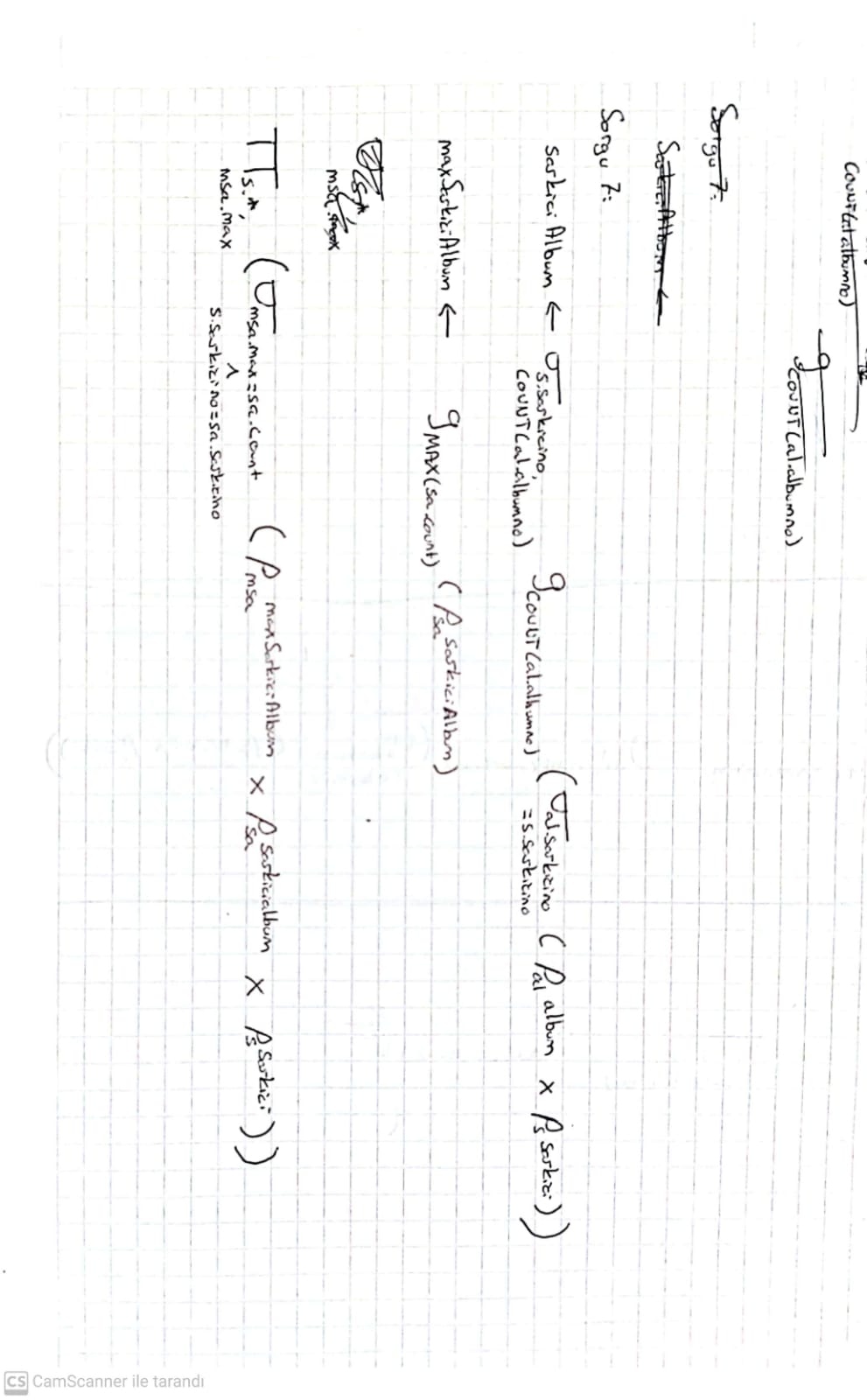
Açıklama otomatik olarak oluşturuldu

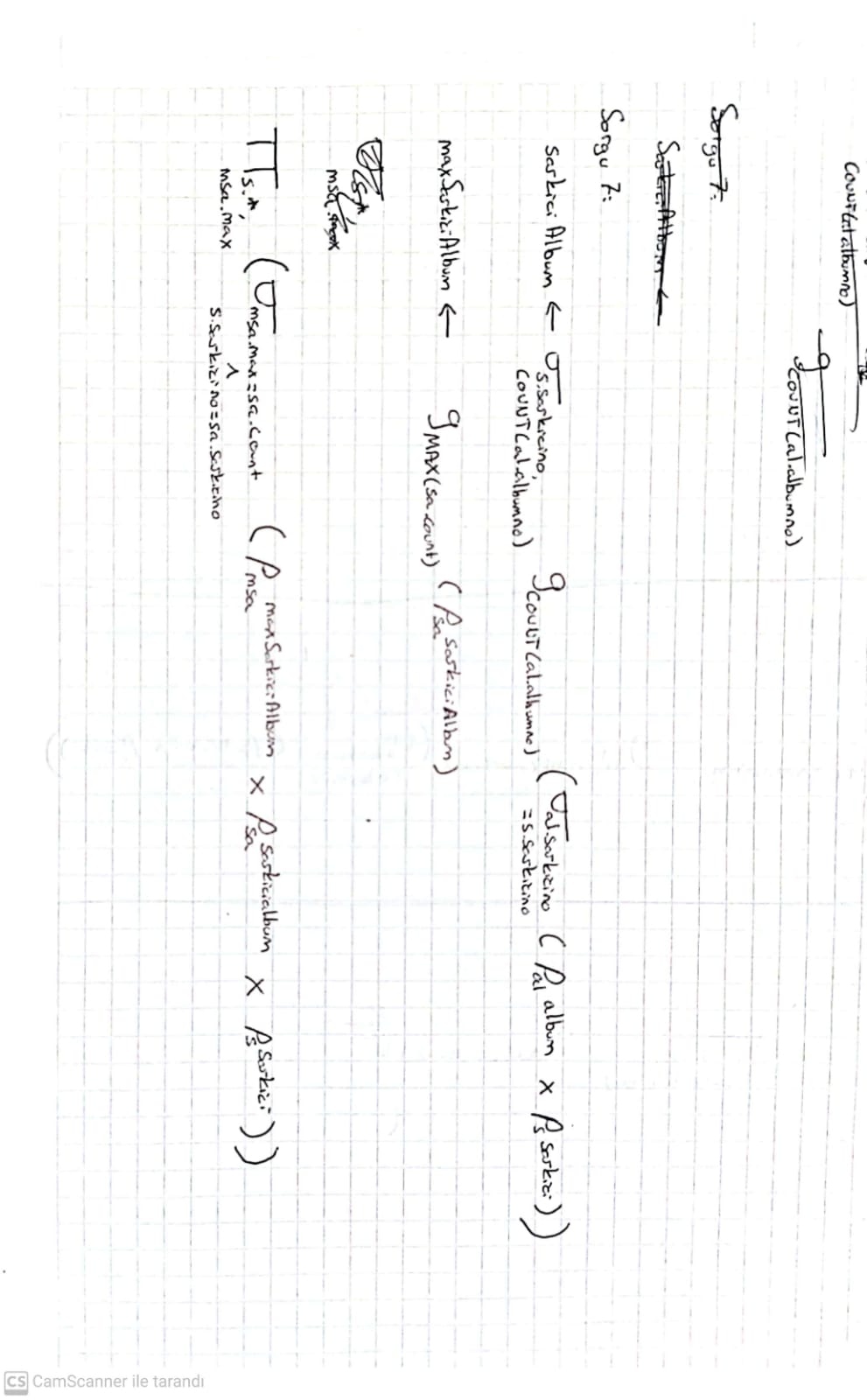


Sorgu 7:

En fazla albüme sahip şarkıcıların kayıtlarını listeleyiniz. [Not:Önce her şarkıcının album sayısı GROUP BY ile bulunur, sonra en yüksek album sayısı bulunur. Sonra en yüksek album sayısına sahip şarkıcıların kayıtları bulunur. Ön adımdaki sorguları WITH ile yazınız]

|  |
| --- |
| WITH  sarkiciAlbum AS (  SELECT s.sarkicino, COUNT(al.albumno) as "count"  FROM album al, sarkici s  WHERE al.sarkicino = s.sarkicino  GROUP BY s.sarkicino  ),  maxSarkiciAlbum AS(  SELECT MAX(sa.count) as "max"  FROM sarkiciAlbum sa  )  SELECT s.\*, msa.max  FROM maxSarkiciAlbum msa,sarkiciAlbum sa, sarkici s  WHERE msa.max = sa.count AND s.sarkicino = sa.sarkicino |



metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, ekran görüntüsü, ekran, kapat içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sorgu 8:

‘arabesk’ türündeki tüm şarkıların yazarlığını yapmış yazarların kayıtlarını listeleyiniz. [Not: İlişkisel cebirdeki bölme (÷) işlemi kullanılacak ve ilk adımda yazarların numaraları bulunduktan sonra ikinci adımda yazar kayıtları bulunacak. Bu sorguyu dersteki *tüm dersleri alan öğrenciler* sorgusuna benzeterek çözebilirsiniz]

|  |
| --- |
| SELECT \*  FROM sozyazari so  WHERE NOT EXISTS (  (  SELECT DISTINCT s.tur  FROM sarki s  WHERE s.yazarno = so.yazarno  )  EXCEPT  (  SELECT DISTINCT s.tur  FROM sarki s  WHERE s.tur = 'Arabesk'  )  ) |

metin, beyaz tahta içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, beyaz tahta içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin, siyah, ekran, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu

Sorgu 9:

En çok şarkı sözü yazmış olan yazarların yazarno’larını aggregate fonksiyon veya group by kullanmadan sınıfta (sunumdaki örnekte) çözdüğümüz gibi çözünüz.

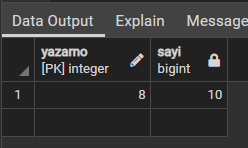
|  |
| --- |
| WITH besteciEserSayisi as (  SELECT so.yazarno as "yazarno", COUNT(so.yazarno) as "sayi"  FROM sozyazari so,sarki s  WHERE so.yazarno = s.yazarno  GROUP BY so.yazarno),  enFazlaEser as(  SELECT MAX(sayi) as "sayi"  FROM besteciEserSayisi  )  SELECT b.yazarno,b.sayi  FROM enFazlaEser e,besteciEserSayisi b  WHERE e.sayi = b.sayi |

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu



Sorgu 10:

En az iki şarkı bestelemiş olan bestecilerin bestecino’larını aynı tabloyu kendisi ile kartezyen çarpım yaparak sınıfta (sunumdaki örnekte) olduğu gibi çözünüz.

|  |
| --- |
| SELECT DISTINCT s1.bestecino  FROM sarki s1, sarki s2  WHERE s1.bestecino = s2.bestecino  GROUP BY s1.bestecino, s2.bestecino  HAVING COUNT(s1.ad) >= 2; |

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

metin içeren bir resim

Açıklama otomatik olarak oluşturuldu

