

BIMU3064

# Veritabanı Yönetim Sistemleri

ÖDEV 2

Abdulkadir Azmanoğlu  
1306130092

- 1) (10puan-IN/EXISTS/=SOME/ALL) 'Ali KURT' adlı doktorun çalıştığı hastanenin kaydını (hospital tablosundaki tüm sütunları) döndüren sorguyu IN, EXISTS ve SOME/ALL ile 3 farklı şekilde veriniz.

**IN:**

```
SELECT hospital.*
FROM hospital
INNER JOIN doctor ON hospital.hid = doctor.hid
WHERE dname IN ('Ali KURT');
```

**EXISTS:**

```
SELECT hospital.*
FROM hospital
WHERE EXISTS (
    SELECT *
    FROM doctor
    WHERE doctor.hid = hospital.hid
    AND dname = 'Ali KURT'
);
```

**SOME/ALL:** : (SOME is a synonym for ANY)

```
SELECT hospital.*
FROM hospital
INNER JOIN doctor ON doctor.hid = hospital.hid
WHERE dname = ANY (
    SELECT dname
    FROM doctor
    WHERE dname = 'Ali KURT'
);
```

hid	hname	city
8	ISTANBUL UNIVERSITESI CERRAHPASA TIP FAKULTESI HASTANESI	ISTANBUL

- 2) (10puan-EXISTS) 'Ali KURT' adlı doktorun çalışMAdığı hastanelerin kayıtlarını (hospital tablosundaki tüm sütunları) döndüren sorguyu NOT IN, NOT EXISTS ve SOME/ALL ile 3 farklı şekilde veriniz.

**NOT IN:**

```
SELECT *  
FROM hospital  
WHERE hid NOT IN (  
    SELECT hospital.hid  
    FROM hospital  
    INNER JOIN doctor ON doctor.hid = hospital.hid  
    WHERE dname = 'Ali KURT'  
);
```

**NOT EXISTS:**

```
SELECT hospital.*  
FROM hospital  
WHERE NOT EXISTS (  
    SELECT *  
    FROM doctor  
    WHERE doctor.hid = hospital.hid  
    AND dname = 'Ali KURT'  
);
```

**SOME/ALL:** (SOME is a synonym for ANY)

```
SELECT *  
FROM hospital  
WHERE hid <> ANY (  
    SELECT hospital.hid  
    FROM hospital  
    INNER JOIN doctor ON doctor.hid = hospital.hid  
    WHERE dname = 'Ali KURT'  
);
```

hid	hname	city
1	RIZE DEVLET HASTANESI	RIZE
2	RIZE ISKOGLU CAYELI DEVLET HASTANESI	RIZE
3	TRABZON OF DEVLET HASTANESI	TRABZON
4	TRABZON NUMUNE EGITIM VE ARASTIRMA HASTANESI	TRABZON
5	ARTVIN DEVLET HASTANESI	ARTVIN
6	ARTVIN BORCKA DEVLET HASTANESI	ARTVIN
7	ISTANBUL USKUDAR DEVLET HASTANESI	ISTANBUL

**3) (10puan-UNION/EXCEPT/INTERSECT)**

- a. Hem 'Biontech' (testType.name alanını kullanınız) hem de 'Moderna' (testType.name alanını kullanınız) testinden Negatif sonuç alan hastaların kayıtlarını INTERSECT ile listeleyiniz.

```

SELECT p.*
FROM patient p
INNER JOIN test t ON p.pid = t.pid
INNER JOIN testType tt ON tt.tid = t.tid
WHERE t.testResult = 'Neg' AND tt.tname = 'Biontech'

INTERSECT

SELECT p.*
FROM patient p
INNER JOIN test t ON p.pid = t.pid
INNER JOIN testType tt ON tt.tid = t.tid
WHERE t.testResult = 'Neg' AND tt.tname = 'Moderna';

```

pid	pname	status
11	Gonul OKUYAN	Neg

**b. Bir Türk (testType.origin='tr') veya Alman (testType.origin='gr') aşısından Positif sonuç alan hastaların kayırlarını UNION ile listeleyiniz.**

```
SELECT p.*
FROM patient p
INNER JOIN test t ON p.pid = t.pid
INNER JOIN testType tt ON tt.tid = t.tid
WHERE t.testResult = 'Pos' AND tt.origin = 'tr'

UNION

SELECT p.*
FROM patient p
INNER JOIN test t ON p.pid = t.pid
INNER JOIN testType tt ON tt.tid = t.tid
WHERE t.testResult = 'Pos' AND tt.tname = 'de';
```

pid	pname	status
2	Yavuz KARACA	Pos
3	Yasemin BAYIK	Pos
12	Mustafa KURU	Pos
15	Arif ATIS	Pos

**c. Hiç test yaptırmamış olan hastaların kayırlarını EXCEPT/MINUS ile listeleyiniz.**

```
SELECT p.*
FROM patient p
EXCEPT
SELECT patient.*
FROM patient
INNER JOIN test t ON t.pid = patient.pid;
```

pid	pname	status
19	Abuziddin TOKAT	NULL
18	Nizamettin SAK	NULL

- 4) **(10puan-GROUP BY-HAVING)** Testleri orijin (testType.origin) ülkelerine göre gruplayarak; sadece doğruluk oranı (testType.accuracy > 80) yüzde 80'in üzerinde olan testler verileri/kayıtları kullanılarak ve 'tr' haricindeki ülkeler için olmak üzere; her ülke için kaç test yapıldığı (test tablosundan bulunur), testler için toplam kaç para ödendiği (testType.price), yapılan testlerin yüzde kaçının positif, yüzde kaçının negatif geldiğini listeleyiniz.

```
WITH Pos_Tests AS (  
    SELECT tt.origin, tt.tname, COUNT(test.tid) AS Test_Count  
    FROM test  
    INNER JOIN testType tt ON test.tid = tt.tid  
    WHERE testResult = 'Pos' AND accuracy > 80 AND tt.origin != 'tr'  
    GROUP BY tt.origin, tt.tname  
) , Neg_Tests AS (  
    SELECT tt.origin, tt.tname, COUNT(test.tid) AS Test_Count  
    FROM test  
    INNER JOIN testType tt ON test.tid = tt.tid  
    WHERE testResult = 'Neg' AND accuracy > 80 AND tt.origin != 'tr'  
    GROUP BY tt.origin, tt.tname  
)  
SELECT tt.origin, tt.tname, COUNT(test.tid) AS Test_Count, SUM(tt.price) AS Total_Price,  
    ROUND(pt.Test_Count * 100.0 / COUNT(test.tid), 1) AS Pos_Ratio,  
    ROUND(nt.Test_Count * 100.0 / COUNT(test.tid), 1) AS Neg_Ratio  
FROM test  
INNER JOIN testType tt ON test.tid = tt.tid  
INNER JOIN Pos_Tests pt ON pt.origin = tt.origin  
INNER JOIN Neg_Tests nt ON nt.origin = tt.origin  
WHERE accuracy > 80 AND tt.origin != 'tr'  
GROUP BY  
tt.origin,  
tt.tname,  
pt.Test_Count,  
nt.Test_Count;
```

origin	tname	test_count	total_price	pos_ratio	neg_ratio
cn	Coronavac	4	1000	50.0	50.0
de	Biontech	11	1650	27.3	72.7
us	Moderna	6	1200	33.3	66.7

5) (10puan-UNIQUE) Aynı isimli testten (testType.name) iki yada daha fazla yaptırmamış (yani isim bakımından farklı testleri yaptırmış) olan hastaların kayıtlarını UNIQUE ile listeleyiniz.

```

SELECT *
FROM patient
WHERE status IS NOT NULL
EXCEPT
SELECT t.pid, p.pname, p.status
FROM test t
INNER JOIN patient p ON p.pid = t.pid
INNER JOIN testType tt ON tt.tid = t.tid
GROUP BY t.pid, t.tid, tt.tname, p.pname, p.status
HAVING COUNT(p.pname) >= 2;

```

pid	pname	status
10	Memure CIVELEK	Neg
6	Havva COLAK	Neg
4	Necati ERTEN	Neg
8	Yasar BILGIN	Neg
1	Abdulkadir AZMANOGLU	Neg
5	Gulsun YILMAZ	Neg
12	Mustafa KURU	Pos
7	Kamuran AKKUS	Pos

16	Gamze KURU	Neg
3	Yasemin BAYIK	Pos
13	Osman CELEN	Neg
14	Ismail AZMAN	Pos
2	Yavuz KARACA	Pos
9	Ayten ARSLAN	Neg

**6) (10puan WITH/FROM altsorgu) Hasta başı ortalama test sayısından daha fazla test yaptıran hastaların pid'lerini listeleyiniz. Bunu yaparken ilk iki adımı yani "hasta başı test sayısını" bir WITH alt sorgusuyla, "bunların ortalamasını" başka bir WITH sorgusuyla yazınız.**

```
WITH Test_Per_Patient AS(
    SELECT pid, COUNT(tid) AS Test
    FROM test
    GROUP BY pid
), AVR_OF_TPP AS (
    SELECT ROUND(AVG(Test_Per_Patient.Test), 1) AS AVR
    FROM Test_Per_Patient
)
SELECT p.pid, tpp.test AS Number_of_Tests, AVR_OF_TPP.AVR AS Average
FROM patient p, Test_Per_Patient tpp, AVR_OF_TPP
WHERE tpp.pid = p.pid
GROUP BY p.pid, tpp.test, AVR_OF_TPP.AVR
HAVING tpp.Test > AVR_OF_TPP.AVR
ORDER BY pid;
```

pid	number_of_tests	average
2	2	1.9
3	3	1.9
7	3	1.9
8	3	1.9



11	4	1.9
14	2	1.9
15	4	1.9
17	2	1.9
20	3	1.9

7) (10puan SELECT alt sorgusu) Hastane kayılarını listeyiniz. Fakat kayıtları listelerken her hastanedeki doktor ve yatan hasta sayılarını SELECT'in içerisine birer adet alt sorgu ekleyerek hesaplatınız. (SELECT \*, (alt-sorgu1) doktor-sayısı, (alt-sorgu2) hasta-sayısı FROM hastane)

```
SELECT *, (
    SELECT COUNT(d.did) AS dCount
    FROM doctor d
    WHERE h.hid = d.hid
    GROUP BY h.hid
) AS number_of_doctors, (
    SELECT COUNT(DISTINCT a.pid) AS iCount
    FROM admitted a
    WHERE h.hid = a.hid
    AND a.entubationDate IS NOT NULL
    GROUP BY h.hid
) AS number_of_inpatient
FROM hospital h
GROUP BY h.hid;
```

hid	hname	city	number_of_doctors	number_of_inpatient
1	RIZE DEVLET HASTANESI	RIZE	2	1
2	RIZE ISKOGLU CAYELI DEVLET HASTANESI	RIZE	2	1
3	TRABZON OF DEVLET HASTANESI	TRABZON	2	2
4	TRABZON NUMUNE EGITIM VE ARASTIRMA HASTANESI	TRABZON	2	

5	ARTVIN DEVLET HASTANESI	ARTVIN	2	
6	ARTVIN BORCKA DEVLET HASTANESI	ARTVIN	2	2
7	ISTANBUL USKUDAR DEVLET HASTANESI	ISTANBUL	2	
8	ISTANBUL UNIVERSITESI CERRAHPASA TIP FAKULTESI HASTANESI	ISTANBUL	3	1

**8) (10puan-HAVING altsorgusu) Uzmanlık alanı ‘dahiliye’ (specialty=‘dahiliye’) olan en az bir (yani herhangi bir) doktora sahip hastaneler için hid ve o hastanede çalışan tüm doktorların sayısını listeleyiniz (HAVING içerisinde söz konusu hastanenin dahiliye uzmanı çalıştırıp çalıştırmadığı kontrolünün yapılması gerekmektedir.)**

```

SELECT h.hid, (
    SELECT COUNT(d.did)
    FROM doctor d
    WHERE h.hid = d.hid
    GROUP BY h.hid
) AS Number_Of_All_Doctors
FROM doctor doc
INNER JOIN hospital h ON h.hid = doc.hid
GROUP BY h.hid, doc.did
HAVING COUNT(doc.did) >= (
    SELECT COUNT(d.did) AS dCount
    FROM doctor d
    INNER JOIN hospital h ON h.hid = d.hid
    WHERE d.did = doc.did
    GROUP BY h.hid, d.specialty
    HAVING d.specialty = 'Dahiliye'
);

```

hid	number_of_all_doctors
8	3
2	2

- 9) (10puan-VIEWS) statistics (date, noOfTests, noOfPositives, noOfDeaths, noOfEntubed) [ istatistik (tarih, gunlukTestSayisi, gunlukPositifSayisi, gunlukOlumSayisi, gunlukEntubeSayisi) ] şeklinde bir view oluşturunuz. Bu view'i oluşturmak için önce test tablosundaki test kayıtlarını date(tarih) alanına göre GROUP BY yaparak günlük positif ve negatif sayıları bulunabilir. Sonra admitted tablosundaki kayıtlar admissionDate alanına göre GROUP BY yapılarak gunlukOlumSayisi, gunlukEntubeSayisi bulunabilir. Son adımda önceki iki adımda elde edilen sonuçlar gruptamanın yapılda tarih alanı üzerinden join edilir. Bazı günlerde test, bazı günlerde de olum/entube olamayacağı düşünülerek bu join işleminin FULL OUTER JOIN olması gerekmektedir.

```
CREATE VIEW statistics_of_pandemic AS
```

```
WITH no_Of_Pos AS (
```

```
    SELECT testDate, COUNT(pid) as c
```

```
    FROM test
```

```
    GROUP BY testDate, testResult
```

```
    HAVING testResult = 'Pos'
```

```
    ORDER BY testDate
```

```
), no_Of_Death AS (
```

```
    SELECT deathDate, COUNT(pid) as c
```

```
    FROM admitted
```

```
    GROUP BY deathDate
```

```
    HAVING deathDate IS NOT NULL
```

```
    ORDER BY deathDate
```

```
), no_Of_Ent AS (
```

```
    SELECT entubationDate, COUNT(pid) as c
```

```
    FROM admitted
```

```
    GROUP BY entubationDate
```

```
    HAVING entubationDate IS NOT NULL
```

```
    ORDER BY entubationDate
```

```
)
```

```
SELECT t.testDate AS date, COUNT(tid) AS noOfTests,
```

```
       nop.c AS noOfPositives, nod.c AS noOfDeaths, noe.c AS noOfEntubed
```

```
FROM test t
```

```
FULL OUTER JOIN no_Of_Pos nop ON nop.testDate = t.testDate
```

```
FULL OUTER JOIN no_Of_Death nod ON nod.deathDate = t.testDate
```

FULL OUTER JOIN no\_Of\_Ent noe ON noe.entubationDate = t.testDate

GROUP BY t.testDate, nop.c, nod.c, noe.c

ORDER BY t.testDate;

date	nooftests	noofpositives	noofdeaths	noofentubed
2020-06-14	1	NULL	NULL	NULL
2020-06-28	2	NULL	NULL	NULL
2020-07-18	2	NULL	NULL	NULL
2020-08-07	1	1	NULL	1
2020-09-12	1	NULL	NULL	NULL
2020-09-13	1	NULL	NULL	NULL
2020-09-19	1	NULL	NULL	NULL
2020-09-20	1	1	NULL	1
2020-09-26	1	NULL	NULL	NULL
2020-09-28	1	NULL	NULL	NULL
2020-11-02	1	NULL	NULL	NULL
2020-11-06	1	NULL	NULL	NULL
2020-11-08	1	NULL	NULL	NULL
2020-11-09	1	1	NULL	1
2020-11-10	1	1	NULL	NULL
2020-11-16	1	1	1	NULL
2020-11-23	1	1	NULL	NULL
2020-11-30	1	1	NULL	1
2020-12-02	1	NULL	NULL	NULL
2020-12-03	2	NULL	NULL	NULL
2020-12-05	2	NULL	NULL	NULL

2020-12-06	1	NULL	NULL	NULL
2020-12-10	3	2	NULL	3
2020-12-14	1	NULL	NULL	NULL
2020-12-17	4	4	2	1

**10) 'Ali KURT' adlı hastanın son testinden sonra test yaptıran hastaların kayıtlarını listeleyiniz.  
(Bir alt sorguda Ali KURT'un test tarihlerini döndürünüz. Hastaların test tarihini bu alt sorgudaki tarihlerle >SOME/ALL operatörünü kullanarak karşılaştırınız)**

```
SELECT DISTINCT p.*
FROM test t
INNER JOIN patient p ON p.pid = t.pid
WHERE t.testDate > ALL (
    SELECT testDate
    FROM test t
    INNER JOIN patient p ON p.pid = t.pid
    WHERE p.pname = 'Ali KURT'
);
```

pid	pname	status
3	Yasemin BAYIK	Pos
5	Gulsun YILMAZ	Neg
7	Kamuran AKKUS	Pos
8	Yasar BILGIN	Neg
9	Ayten ARSLAN	Neg
10	Memure CIVELEK	Neg
13	Osman CELEN	Neg
14	Ismail AZMAN	Pos
15	Arif ATIS	Pos
20	Gazanfer OZCAN	Pos

