



# Lab 8 QUERIES

202012001-04\_202018001

**JANU SHAH** 202012001

**DEEPAK MAIDASANI** 202012002

**KUSH MODI** 202012003

**NIHARIKA SRIVASTAVA** 202012004

**KUNAL PANJWANI** 202018001

SET SEARCH\_PATH TO online\_medical\_consultation\_system;

### **QUERIES:**

1. For a patient 'XYZ', retrieve doctor's diagnosis, digital prescription and lab reports.

### **RELATIONAL ALGEBRA:**

$$\Pi_{p.name, pr.doctorDiagnosis, pr.labReport, pr.digitalPrescription} (\rho (pr, patientRecord) \bowtie_{\langle patientRecords.patientId=p.patientId \rangle} ( \rho (p, \sigma_{firstName='xyz'} (patient))))$$

### **SQL QUERY:**

```
SELECT p.firstName || ' ' || p.lastName AS name, pr.doctorDiagnosis,
pr.labReport, pr.digitalPrescription
FROM patientRecord AS pr
JOIN (SELECT * FROM patient WHERE upper(firstName) = 'JANU') AS p
ON p.patientId = pr.patientId ;
```

Dashboard

Properties

SQL

Statistics

Dependencies

Dependents

postgres/postgres@PostgreSQL 12 \*

- Retrieve the patient details and recorded video consultation for a given appointment Id 'A05'.

### RELATIONAL ALGEBRA:

$$\Pi_{\text{name, p.dateOfBirth, p.gender, p.mobileNumber, m.videoConferencingLink}} (\rho (\text{m, meetDetails}) \bowtie_{\langle \text{m.appointmentId}=\text{a.appointmentId} \rangle} (\rho (\text{a, } \sigma_{\text{appointmentId}='A05'} (\text{appointment}))) \bowtie_{\langle \text{a.patientId}=\text{p.patientId} \rangle} \rho (\text{p, patient}) )$$

### SQL QUERY:

```

SELECT p.firstName || ' ' || p.lastName AS name, p.dateOfBirth, p.gender,
p.mobileNumber, m.videoConferencingLink
FROM (SELECT * FROM appointment WHERE appointmentId = 'A05') AS a
JOIN patient AS p ON p.patientId = a.patientId
JOIN meetDetails AS m ON m.appointmentId = a.appointmentId

```

Dashboard Properties SQL Statistics Dependencies Dependents postgres/postgres@PostgreSQL 12 \*

postgres/postgres@PostgreSQL 12

Query Editor Query History

```

1041 SET SEARCH_PATH TO online_medical_consultation_system;
1042 SELECT p.firstName || ' ' || p.lastName AS name, p.dateOfBirth, p.gender, p.mobileNumber, m.video
1043 FROM (SELECT * FROM appointment WHERE appointmentId = 'A05') AS a
1044 JOIN patient AS p ON p.patientId = a.patientId
1045 JOIN meetDetails AS m ON m.appointmentId = a.appointmentId
1046
1047
1048
1049
1050
1051
1052
1053

```

Data Output Explain Messages Notifications

	name text	dateofbirth date	gender character (1)	mobilenumber numeric (10)	videoconferencinglink character varying (50)
1	swati ...	1973-10-17	F	9702461510	

- List all doctors from 'MB07' medical branch working at 'HB05' branch of the hospital.

### RELATIONAL ALGEBRA:

$\Pi_{\text{name}}(\sigma_{\text{branchId}='HB05' \text{ AND } \text{medicalBranchId}='MB07'}(\text{doctor}))$

### SQL QUERY:

SELECT firstName || ' ' || lastName AS name

FROM doctor

WHERE branchId = 'HB05' AND medicalBranchId = 'MB07'

postgres/postgres@PostgreSQL 12

Query Editor   Query History

```

11 Set search_path to online_medical_consultation_system;
12
13 SELECT firstName || ' ' || lastName AS name
14 FROM doctor
15 WHERE branchId = 'HB05' AND medicalBranchId = 'MB07'
16

```

Data Output   Explain   Messages   Notifications

	name text	
1	Niharika Oberoi	
2	Deepak Maidashani	
3	Ankita Oberoi	

- List all the patient details who booked an appointment since 1st March, 2020 .

#### RELATIONAL ALGEBRA:

$$\Pi_{\text{name}} ((\rho(a, \sigma_{\text{dateOfAppointment} \geq '2020-03-01' \text{ AND } \text{dateOfAppointment} \leq \text{CURRENT\_DATE}(\text{appointment})) \bowtie_{\langle a.\text{patientId}=p.\text{patientId} \rangle} \rho(p, \text{patient})))$$

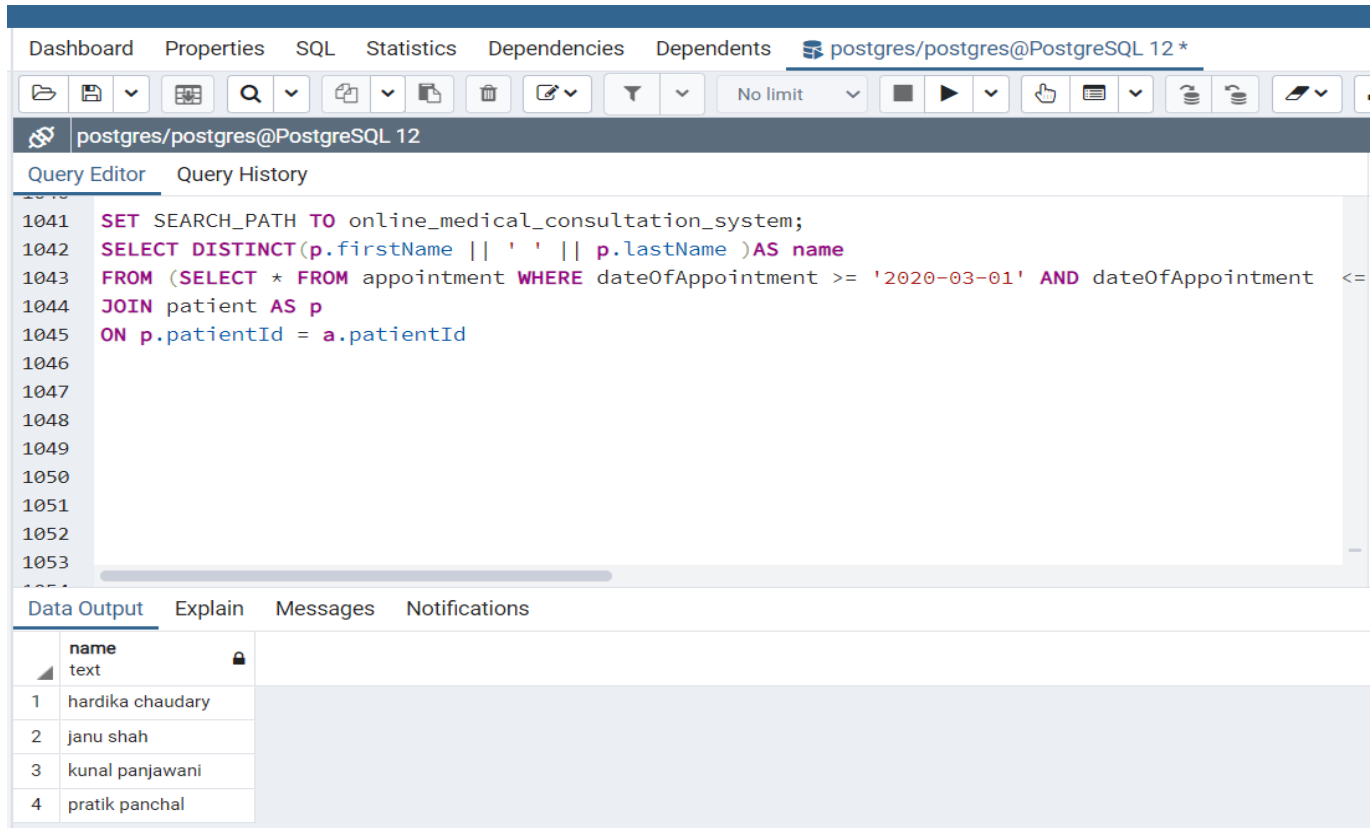
#### SQL QUERY:

```

SELECT DISTINCT(p.firstName || ' ' || p.lastName )AS name
FROM (SELECT * FROM appointment WHERE dateOfAppointment >=
'2020-03-01' AND dateOfAppointment <= CURRENT_DATE) AS a

```

JOIN patient AS p  
ON p.patientId = a.patientId



The screenshot shows a PostgreSQL query editor interface. The query editor displays the following SQL query:

```

1041 SET SEARCH_PATH TO online_medical_consultation_system;
1042 SELECT DISTINCT(p.firstName || ' ' || p.lastName )AS name
1043 FROM (SELECT * FROM appointment WHERE dateOfAppointment >= '2020-03-01' AND dateOfAppointment <=
1044 JOIN patient AS p
1045 ON p.patientId = a.patientId
1046
1047
1048
1049
1050
1051
1052
1053

```

Below the query editor, the "Data Output" tab is active, showing the results of the query. The results are displayed in a table with the following columns: "name" (text) and a lock icon. The table contains four rows of data:

	name
1	hardika chaudary
2	janu shah
3	kunal panjawani
4	pratik panchal

5. List the number of consultations held for a patient 'xyz' in the year 2019 and 2020.

### RELATIONAL ALGEBRA:

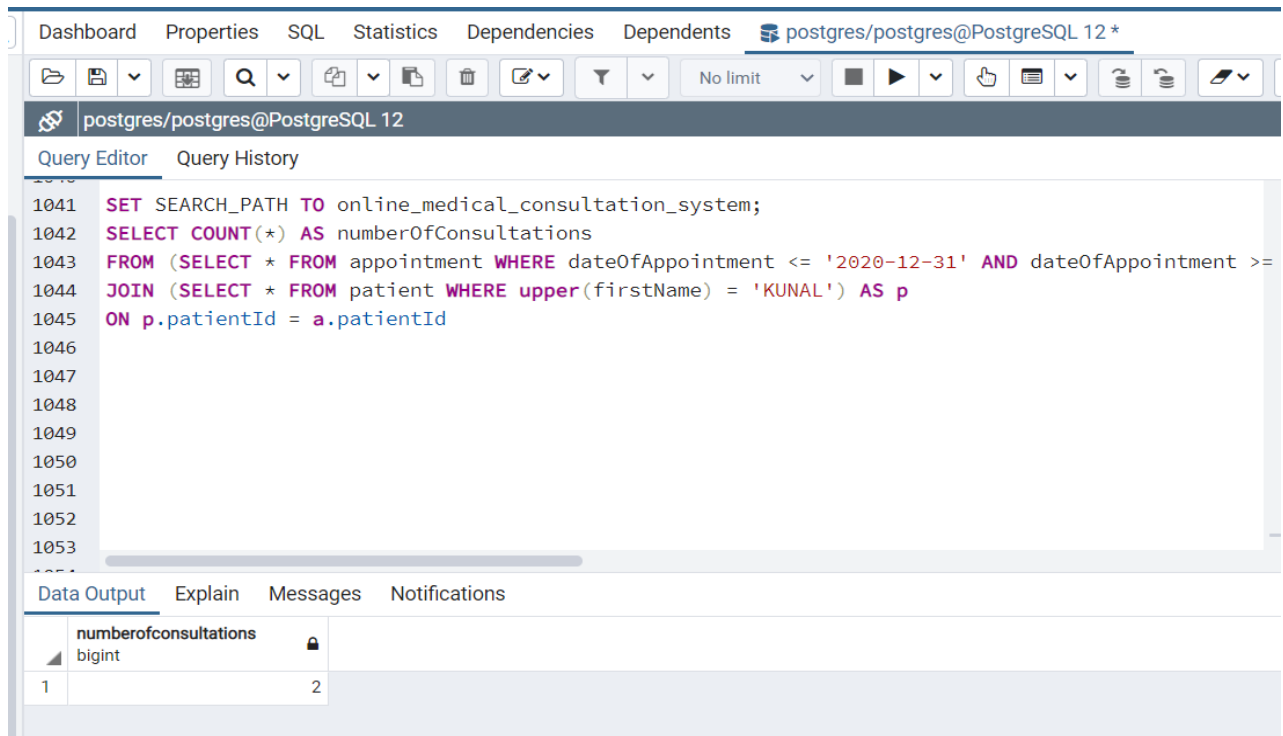
$\mathcal{F}COUNT(*) \rightarrow \text{numberOfConsultations} ((\rho(a, \sigma_{\text{dateOfAppointment} \geq '2018-01-01' \text{ AND } \text{dateOfAppointment} \leq '2020-12-31'}(\text{appointment}))) \bowtie_{\langle a.\text{patientId} = p.\text{patientId} \rangle} \rho(p, \sigma_{\text{firstName} = 'KUNAL'}(\text{patient})))$

### SQL QUERY:

```

SELECT COUNT(*) AS numberOfConsultations
FROM (SELECT * FROM appointment WHERE dateOfAppointment <=
'2020-12-31' AND dateOfAppointment >= '2018-01-01') AS a
JOIN (SELECT * FROM patient WHERE upper(firstName) = 'KUNAL') AS p
ON p.patientId = a.patientId

```



- List names of doctors from 'D' Medical Branch who completed the maximum number of consultations between the years 2010 and 2020.

### RELATIONAL ALGEBRA:

$r1 \leftarrow \rho(a, \sigma_{\text{dateOfAppointment} \geq '2020-01-01' \text{ AND } \text{dateOfAppointment} \leq '2020-12-31'}(\text{appointment}))$

$\bowtie_{<d.\text{doctorId} = a.\text{doctorId} >} \rho(d, \text{doctor})$

$\bowtie_{<mb.medicalBranchId = d.medicalBranchId>} \rho(mb, \sigma_{UPPER(medicalBranchName) = 'CARDIOLOGY'}(medicalBranch))$

$r2 \leftarrow d.doctorId \mathcal{F}_{name}, COUNT(a.appointmentId) \rightarrow cnt \text{ (r1)}$

$r3 \leftarrow \mathcal{F}_{MAX(cnt)} \rightarrow maxcnt \text{ (r2)}$

$r4 \leftarrow \sigma_{cnt = cnt} (r3 \times r2)$

$result \leftarrow \Pi_{name}(r4)$

### SQL QUERY:

SELECT details.name

FROM

(

SELECT MAX(cnt) AS cnt

FROM

(

SELECT COUNT(a.appointmentId) AS cnt

FROM (SELECT \* FROM appointment WHERE

dateOfAppointment <= '2020-12-31' AND dateOfAppointment >= '2010-01-01') AS a

JOIN doctor AS d

ON a.doctorId = d.doctorId

JOIN (SELECT medicalBranchId from medicalBranch WHERE UPPER(medicalBranchName) ='CARDIOLOGY') AS mb

ON mb.medicalBranchId = d.medicalBranchId

GROUP BY d.doctorId

) AS countOfAppts

) AS maxCount

CROSS JOIN



```

(
    SELECT d.firstName || ' ' || d.lastName AS name,
    COUNT(a.appointmentId) AS cnt

    FROM (SELECT * FROM appointment WHERE dateOfAppointment <=
'2020-12-31' AND dateOfAppointment >= '2010-01-01') AS a

    JOIN doctor AS d

    ON a.doctorId = d.doctorId

    JOIN (SELECT medicalBranchId from medicalBranch WHERE
UPPER(medicalBranchName)='CARDIOLOGY') AS mb

    ON mb.medicalBranchId = d.medicalBranchId

    GROUP BY d.doctorId

) AS details
WHERE details.cnt = maxCount.cnt

```

The screenshot shows the pgAdmin 4 interface. The Query Editor contains the following SQL query:

```

1055 SET SEARCH_PATH TO online_medical_consultation_system;
1056 SELECT details.name FROM
1057 (SELECT MAX(cnt) AS cnt
1058 FROM
1059 ( SELECT COUNT(a.appointmentId) AS cnt
1060 FROM (SELECT * FROM appointment WHERE dateOfAppointment <= '2020-12-31' AND dateOfAppointment >= '2010-01-01') AS a
1061 JOIN doctor AS d
1062 ON a.doctorId = d.doctorId
1063 JOIN (SELECT medicalBranchId from medicalBranch WHERE UPPER(medicalBranchName)='CARDIOLOGY') AS mb
1064 ON mb.medicalBranchId = d.medicalBranchId
1065 GROUP BY d.doctorId
1066 ) AS countOfAppts
1067 ) AS maxCount
1068 CROSS JOIN
1069 (SELECT d.firstName || ' ' || d.lastName AS name, COUNT(a.appointmentId) AS cnt
1070 FROM (SELECT * FROM appointment WHERE dateOfAppointment <= '2020-12-31' AND dateOfAppointment >= '2010-01-01') AS a
1071 JOIN doctor AS d
1072 ON a.doctorId = d.doctorId
1073 JOIN (SELECT medicalBranchId from medicalBranch WHERE UPPER(medicalBranchName)='CARDIOLOGY') AS mb
1074 ON mb.medicalBranchId = d.medicalBranchId
1075 GROUP BY d.doctorId
1076 ) AS details
1077 WHERE details.cnt = maxCount.cnt
1078

```

The Data Output pane shows the following result:

name
1 Sandeep Vaishya

7. List the hospital branch with the least number of consultations in 2020.

RELATIONAL ALGEBRA:

$r1 \leftarrow \rho(d, \text{doctor})$

$\bowtie_{\langle d.\text{branchId} = h.\text{branchId} \rangle} \rho(h, \text{hospitalBranch})$

$\bowtie_{\langle d.\text{doctorId} = a.\text{doctorId} \rangle} \rho(a, \text{appointment})$

$r2 \leftarrow \pi_{d.\text{branchId}} \sigma_{d.\text{branchId} \rightarrow \text{branch}, \text{COUNT}(a.\text{appointmentId}) \rightarrow \text{cnt}}(r1)$

$r3 \leftarrow \sigma_{\text{MIN}(\text{cnt}) \rightarrow \text{mincnt}}(r2)$

$r4 \leftarrow \sigma_{\text{cnt} = \text{mincnt}}(r3 \times r2)$

$\text{result} \leftarrow \pi_{\text{branch}}(r4)$

SQL QUERY:

SELECT cntBranch.branch

FROM

(

SELECT d.branchId AS branch, COUNT(a.appointmentId) AS

cnt

FROM doctor AS d

JOIN hospitalBranch AS h ON h.branchId = d.branchId

JOIN appointment AS a ON a.doctorId = d.doctorId

GROUP BY d.branchId

) AS cntBranch

CROSS JOIN

( SELECT MIN(cnt) AS mincnt

FROM

```

(
    SELECT d.branchId AS branch, COUNT(a.appointmentId)
AS cnt
    FROM doctor AS d
    JOIN hospitalBranch AS h ON h.branchId = d.branchId
    JOIN appointment AS a ON a.doctorId = d.doctorId
    GROUP BY d.branchId
) AS cntBranch
) AS minCount
WHERE cntBranch.cnt = mincnt
ORDER BY cntBranch.branch

```

The screenshot shows the PgAdmin interface with a SQL query executed in the Query Editor. The query is as follows:

```

1055 SET SEARCH_PATH TO online_medical_consultation_system;
1056 SELECT cntBranch.branch FROM
1057     (SELECT d.branchId AS branch, COUNT(a.appointmentId) AS cnt
1058      FROM doctor AS d
1059      JOIN hospitalBranch AS h ON h.branchId = d.branchId
1060      JOIN appointment AS a ON a.doctorId = d.doctorId
1061      GROUP BY d.branchId
1062     ) AS cntBranch
1063 CROSS JOIN
1064     ( SELECT MIN(cnt) AS mincnt FROM
1065       ( SELECT d.branchId AS branch, COUNT(a.appointmentId) AS cnt
1066        FROM doctor AS d
1067        JOIN hospitalBranch AS h ON h.branchId = d.branchId
1068        JOIN appointment AS a ON a.doctorId = d.doctorId
1069        GROUP BY d.branchId
1070       ) AS cntBranch
1071     ) AS minCount
1072 WHERE cntBranch.cnt = mincnt
1073 ORDER BY cntBranch.branch

```

The results are displayed in the Data Output tab, showing a table with the following data:

	branch
1	HB02
2	HB07

8. Retrieve the patient details who booked a specific lab Id 'L01' in 'HB01' hospital branch for a pathology test at home.

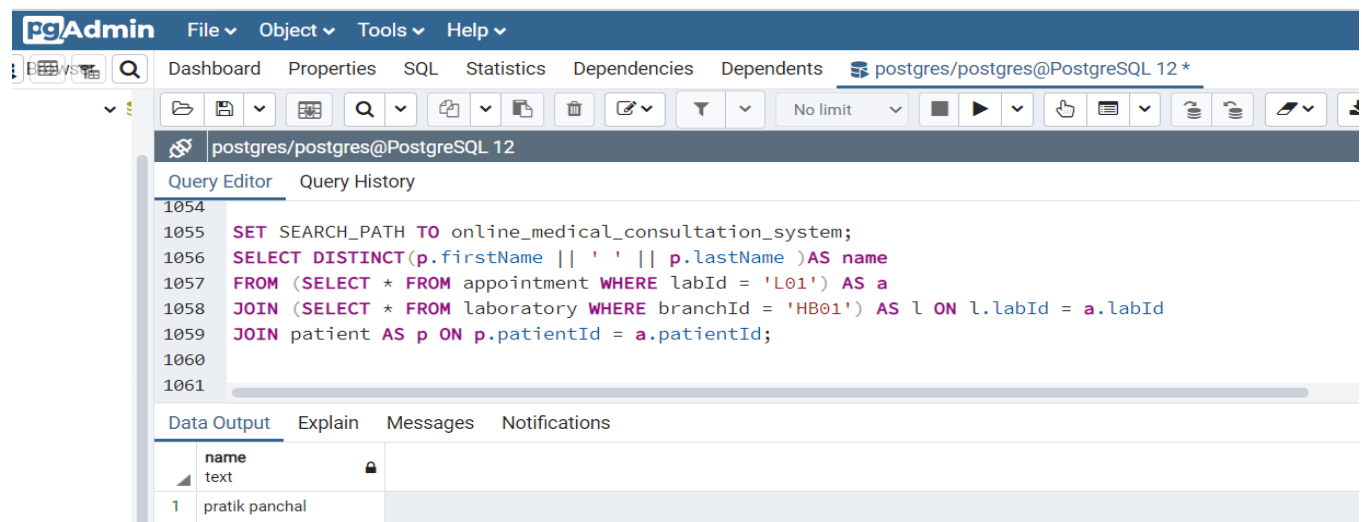
### RELATIONAL ALGEBRA:

$\Pi_{\text{name}} ((\rho(a, \sigma_{\text{labId} = 'L01'}(\text{appointment}))) \bowtie_{\langle a.\text{labId} = l.\text{labId} \rangle}$

$\rho(l, \sigma_{\text{branchId} = 'HB01'}(\text{laboratory})) \bowtie_{\langle a.\text{patientId} = p.\text{patientId} \rangle} \rho(p, \text{patient})$

### SQL QUERY:

```
SELECT DISTINCT(p.firstName || ' ' || p.lastName )AS name
FROM (SELECT * FROM appointment WHERE labId = 'L01') AS a
JOIN (SELECT * FROM laboratory WHERE branchId = 'HB01') AS l ON
l.labId = a.labId
JOIN patient AS p ON p.patientId = a.patientId
```



9. List the doctors with consultation charges more than the average consultation charges.

## RELATIONAL ALGEBRA:

$\Pi_{\text{name}} \sigma_{\text{chargePerConsultation} \geq (\mathcal{F}\text{AVG}(\text{chargePerConsultation}) (\text{doctor}))} (\text{doctor})$

## SQL QUERY:

```
SELECT firstName || ' ' || lastName AS name
FROM doctor
WHERE chargePerConsultation >=
(SELECT AVG(chargePerConsultation)
FROM doctor)
```

Query Editor		Query History																		
31	Set search_path to online_medical_consultation_system;																			
32																				
33	SELECT firstName    ' '    lastName AS name																			
34	FROM doctor																			
35	WHERE chargePerConsultation >=																			
36	(SELECT AVG(chargePerConsultation)																			
37	FROM doctor)																			
Data Output		Explain Messages Notifications																		
	<table><thead><tr><th></th><th>name</th><th></th></tr><tr><th></th><th>text</th><th></th></tr></thead><tbody><tr><td>1</td><td>Naresh Trehan</td><td></td></tr><tr><td>2</td><td>Sandeep Vaishya</td><td></td></tr><tr><td>3</td><td>Deepak Maidashani</td><td></td></tr><tr><td>4</td><td>Manan Seth</td><td></td></tr></tbody></table>		name			text		1	Naresh Trehan		2	Sandeep Vaishya		3	Deepak Maidashani		4	Manan Seth		
	name																			
	text																			
1	Naresh Trehan																			
2	Sandeep Vaishya																			
3	Deepak Maidashani																			
4	Manan Seth																			

10. Retrieve the patients who have paid more than Rs. 5000 in the year 2020 for their online consultation.

## RELATIONAL ALGEBRA:

$r1 \leftarrow \rho(a, \sigma_{\text{dateOfAppointment} \geq '2020-01-01' \text{ AND } \text{dateOfAppointment} \leq '2020-12-31'} (\text{appointment}))$

$\bowtie_{\langle a.\text{appointmentId}=p.\text{appointmentId} \rangle} \rho(\text{py}, \text{payment})$

$\bowtie_{\langle a.\text{patientId}=p.\text{patientId} \rangle} \rho(p, \text{patient})$

$r2 \leftarrow a.\text{patientId} \mathcal{F}_{a.\text{patientId}}(r1)$

$r3 \leftarrow \sigma_{\text{SUM}(\text{py.transactionAmount}) > 5000}(r2)$

$r4 \leftarrow \Pi_{\text{patientId}}(r3)$

$\text{result} \leftarrow \text{patient SEMI-INTERSECTION}_{\langle \text{patientId} = \text{patientId} \rangle}(r4)$

SQL QUERY:

SELECT firstName || ' ' || lastName AS name

FROM patient

WHERE patientId IN

(

SELECT a.patientId

FROM (SELECT \* FROM appointment WHERE dateOfAppointment >= '2020-01-01' AND dateOfAppointment <= '2020-12-31') AS a

JOIN payment AS py

ON a.appointmentId = py.appointmentId

JOIN patient AS p ON p.patientId = a.patientId

GROUP BY a.patientId

HAVING SUM(py.transactionAmount) > 5000

)

```

40
41 SELECT firstName || ' ' || lastName AS name
42 FROM patient
43 WHERE patientId IN
44 (
45 SELECT a.patientId
46 FROM (SELECT * FROM appointment WHERE dateOfAppointment >= '2020-01-01' AND dateOfAppointment <= '2020-12-31') AS a
47 JOIN payment AS py
48 ON a.appointmentId = py.appointmentId
49 JOIN patient AS p ON p.patientId = a.patientId
50 GROUP BY a.patientId
51 HAVING SUM(py.transactionAmount) > 5000
52 )

```

Data Output Explain Messages Notifications

	name	
	text	
1	hardika chaudary	
2	swati mehta	
3	arvind bhandari	

11. List the doctor details who has been booked for the second highest number of appointments.

RELATIONAL ALGEBRA:

$r1 \leftarrow \text{doctorid} \mathcal{F}_{\text{count}(*)->\text{cnt}(\text{Appointment})} \text{limit}(1)$

$r2 \leftarrow \text{doctorid} \mathcal{F}_{\text{doctorid}, \text{count}(*)->\text{cnt}(\text{cnt}<r1)} \text{limit}(1)$

$\text{result} \leftarrow \sigma(\text{doctor} \bowtie_{<\text{doctor.doctorid}=r2.doctorid>} r2)$

SQL QUERY:

```

SELECT * FROM doctor
AS d
JOIN
(
    SELECT doctorid,count(*) as cnt
    FROM appointment GROUP BY doctorid
    HAVING count(*)<(SELECT count(*)
    AS cnt FROM appointment
    GROUP BY doctorid ORDER BY cnt desc limit 1)
    ORDER BY cnt desc limit 1
) AS apt ON d.doctorid=apt.doctorid;

```



The screenshot shows a PostgreSQL query editor with the following SQL query:

```

1 SELECT * FROM doctor
2 AS d
3 JOIN
4 (
5     SELECT doctorid,count(*) as cnt
6     FROM appointment GROUP BY doctorid
7     HAVING count(*)<(SELECT count(*)
8     AS cnt FROM appointment
9     GROUP BY doctorid ORDER BY cnt desc limit 1)
10    ORDER BY cnt desc limit 1
11 ) AS apt ON d.doctorid=apt.doctorid;
12

```

The results pane shows a single row of data:

doctorid	character varying (5)	firstname	character varying (20)	lastname	character varying (20)	qualification	character varying (20)	specialization	character varying (20)	experience	numeric (2)	dateofbirth	date	gender	character (1)	mob	num
1	D08	Kush	Shah	MBBS	ENT	21	1975-05-15	M									

12. List the patient details who have ordered medicine from the online pharmacy in 2020.

### RELATIONAL ALGEBRA:

$$r1 \leftarrow \sigma_{\langle \text{patientrecord.patientid} = \text{pharmacy.pharmacyid} \rangle} (\text{patientrecord} \bowtie \text{pharmacy})$$

$$\text{result} \leftarrow \Pi_{\text{patient.name}, \text{patient.lastname}, \text{patient.dateOfBirth}, \text{patient.gender}, \text{patient.mobileNumber}} (\text{patient} \bowtie_{\langle \text{patient.patientid} = r1.patientid \rangle} r1)$$

### SQL QUERY:



```

SELECT pat.firstName || ' ' || pat.lastName AS name,
pat.dateOfBirth, pat.gender, pat.mobileNumber
FROM patient AS pat
JOIN
(
    SELECT * FROM patientrecord AS pr
    JOIN pharmacy AS ph ON pr.pharmacyid=ph.pharmacyid
) AS rec
ON pat.patientid=rec.patientid;

```

Query Editor		Query History	
80	Set search_path to online_medical_consultation_system;		
81	SELECT pat.firstName    ' '    pat.lastName AS name,		
82	pat.dateOfBirth, pat.gender, pat.mobileNumber		
83	FROM patient AS pat		
84	JOIN		
85	(		
86	SELECT * FROM patientrecord AS pr		
87	JOIN pharmacy AS ph ON pr.pharmacyid=ph.pharmacyid		
88	) AS rec		
89	ON pat.patientid=rec.patientid;		
90			

Data Output		Explain	Messages	Notifications
	name text	dateofbirth date	gender character (1)	mobilenumber numeric (10)
1	kush modi	1999-01-01	M	9833274471
2	hardika chaudary	2000-03-24	F	7654321098
3	pratik panchal	1976-08-14	M	8849424642
4	sonal bhandari	1997-10-01	F	1298347645
5	deepak gupta	2002-11-10	M	981234567
6	janu shah	2000-08-23	F	1236547890
7	kunal panjawani	1999-05-02	M	9876543210
8	janu shah	2000-08-23	F	1236547890
9	manan mehta	1973-01-17	M	9701273710
10	arvind bhandari	1997-06-01	M	1296785445
11	ankita gupta	2002-12-10	F	989807667

13.Retrieve the reviews given by a patient 'P' for a doctor 'D'.

### RELATIONAL ALGEBRA:

$r1 \leftarrow \rho(a, \text{appointment})$

$\bowtie_{\langle a.\text{patientId} = p.\text{patientId} \rangle} \rho(p, \sigma_{\text{patientname}='kunal'}(\text{patient}))$

$\bowtie_{\langle a.\text{doctorId} = d.\text{doctorId} \rangle} \rho(d, \sigma_{\text{firstname}='Janu'}(\text{doctor}))$

$r2 \leftarrow \Pi_{a.\text{appointmentid}}(r1)$

```
r3<- feedback SEMI-INTERSECTIONfeedback.appointmentid=a.appointmentid(r2)
result<-  $\Pi_{\text{reviewbypatient}}$ (r3)
```

### SQL QUERY:

```
SELECT reviewByPatient
```

```
FROM feedback
```

```
WHERE appointmentId IN
```

```
(
```

```
    SELECT a.appointmentId
```

```
    FROM appointment AS a
```

```
    JOIN (SELECT * FROM patient WHERE firstName ='kunal') AS p
```

```
    ON a.patientId = p.patientId
```

```
    JOIN (SELECT * FROM doctor WHERE firstName = 'Janu') AS d
```

```
    ON a.doctorId = d.doctorId
```

```
)
```

Query Editor

Query History

```

99  Set search_path to online_medical_consultation_system;
100
101  SELECT reviewByPatient
102  FROM feedback
103  WHERE appointmentId IN
104  (
105      SELECT a.appointmentId
106      FROM appointment AS a
107      JOIN (SELECT * FROM patient WHERE firstName ='kunal') AS p
108      ON a.patientId = p.patientId
109      JOIN (SELECT * FROM doctor WHERE firstName = 'Janu') AS d
110      ON a.doctorId = d.doctorId
111  )

```

Data Output

Explain

Messages

Notifications

	reviewbypatient character varying (100)	
1	no result after taking medicines	
2	no effect even after 4 weeks	

14. List the doctor who has the lowest rating in the 'X' medical branch.

RELATIONAL ALGEBRA:

$r1 \leftarrow \Pi_{\text{medicalbranchid}}(\sigma_{\text{medicalbranchname}='Cardiology'}(\text{medicalbranch}))$

$r2 \leftarrow \text{doctor SEMI-INTERSECTION}_{\text{doctor.medicalbranchid}=\text{medicalbranch.medicalbranchid}}(r1)$

$r3 \leftarrow \mathcal{F}_{\text{MIN}(\text{rating}) \rightarrow \text{minrating}}(r2)$

$r4 \leftarrow \Pi_{\text{firstname}, \text{rating}}(r2)$

$r5 \leftarrow \sigma_{\text{rating}=\text{minrating}}(r4 \times r3)$

$\text{result} \leftarrow \Pi_{\text{firstname}}(r5)$

SQL QUERY:

SELECT concat(docRating.firstName, ' ', docRating.lastName), rating  
FROM

(  
    SELECT firstName, lastName, rating  
    FROM doctor  
    WHERE medicalBranchId IN  
    (  
        SELECT medicalBranchId FROM medicalBranch  
        WHERE medicalBranchName ='Cardiology'  
    )  
) AS docRating

CROSS JOIN

(  
    SELECT MIN(rating) as minrating  
    FROM doctor  
    WHERE medicalBranchId IN  
    (  
        SELECT medicalBranchId FROM medicalBranch  
        WHERE medicalBranchName ='Cardiology'  
    )  
)

) AS docMinRating  
 WHERE docRating.rating = docMinRating.minrating

Query Editor

Query History

```

271 SELECT concat(docRating.firstName,' ',docRating.lastName),rating
272 FROM
273 (
274 SELECT firstName, lastName,rating
275 FROM doctor
276 WHERE medicalBranchId IN
277 (
278 SELECT medicalBranchId FROM medicalBranch
279 WHERE medicalBranchName ='Cardiology'
280 )
281 ) AS docRating
282 CROSS JOIN
283 (
284 SELECT MIN(rating) as minrating
285 FROM doctor
286 WHERE medicalBranchId IN
287 (
288 SELECT medicalBranchId FROM medicalBranch
289 WHERE medicalBranchName ='Cardiology'
290 )
291 ) AS docMinRating
292 WHERE docRating.rating = docMinRating.minrating

```

Data Output

Explain

Messages

Notifications

	concat text	rating numeric (1)	
1	Sandeep Vaishya	1	
2	Arvind Vaishya	1	

15.List the hospital branches located in ‘X’, ‘Y’, or ‘Z’ cities.

### RELATIONAL ALGEBRA:

$\Pi_{branchid}(\sigma_{\langle cityname='Ahmedabad' \text{ OR } cityname='Patna' \text{ OR } cityname='Chandigarh' \rangle}(\text{hospitalbranch} \bowtie_{\langle hospitalbranch.cityid=city.cityid \rangle} city))$

### SQL QUERY:

SELECT branchid  
 FROM hospitalbranch

JOIN city  
 ON (hospitalbranch.cityid=city.cityid)  
 WHERE cityname='Ahmedabad' OR cityname='Patna' OR  
 cityname='Chandigarh'

postgres/postgres@PostgreSQL 13

Query Editor Query History

```

1 SELECT branchid
2 FROM hospitalbranch
3 JOIN city
4 ON (hospitalbranch.cityid=city.cityid)
5 WHERE cityname='Ahmedabad' OR cityname='Patna' OR cityname='Chandigarh'
6

```

Data Output Explain Messages Notifications

	branchid
1	HB01
2	HB04
3	HB05
4	HB09

16. List the appointments having appointment status as 'cancelled' for a doctor 'A'.

### RELATIONAL ALGEBRA:

$\sigma_{\text{doctorid}='D03' \text{ AND } \text{appointmentstatus}='cancelled'}(\text{Appointment})$

### SQL QUERY:

SELECT \* FROM appointment WHERE  
 doctorid='D03' AND appointmentstatus='cancelled'

postgres/postgres@PostgreSQL 12								
Query Editor   Query History								Scratch Pad ✕
<pre> 127 Set search_path to online_medical_consultation_system; 128 129 SELECT * FROM appointment WHERE 130 doctorid='D03' AND appointmentstatus='cancelled' 131 132 </pre>								
Data Output   Explain   Messages   Notifications								
	appointmentid [PK] character varying (10)	dateofappointment date	timeofappointment time without time zone	symptoms character varying (100)	appointmentstatus character varying (16)	durationofappointment time without time zone	doctorid character varying (5)	patientid character varying (10)
1	A03	2018-01-09	11:30:00	fever	cancelled	02:00:00	D03	P03
2	A29	2018-12-09	07:20:00	headache	cancelled	00:00:00	D03	P13

17. List the top 3 labs which have been booked the maximum number of times in the year 2020.

### RELATIONAL ALGEBRA:

l.labname  $\mathcal{F}$ COUNT(\*) -> numberofconsultations

(( $\rho$  (a,  $\sigma_{\text{dateOfAppointment} \geq '2020-01-01'}$

AND  $\text{dateOfAppointment} \leq '2020-12-31'$  (appointment)))

$\bowtie_{<\text{abc.labid}=\text{l.labId}> \mathcal{F}\text{labname}}$  )

### SQL QUERY:

```

SELECT l.labname,
COUNT(*) AS maxnumberofbookings
FROM
(
SELECT * FROM appointment
WHERE dateofappointment>='2020-01-01'
AND dateofappointment<='2020-12-31'
)
AS abc
JOIN
(
SELECT * FROM laboratory
) AS l
ON l.labid=abc.labid GROUP BY labname
ORDER BY count(*)
LIMIT 3

```

postgres/postgres@PostgreSQL 13

Query Editor Query History

```

1 SET SEARCH_PATH TO online_medical_consultation_system;
2
3 SELECT l.labname,
4 COUNT(*) AS maxnumberofbookings
5 FROM
6 (
7 SELECT * FROM appointment
8 WHERE dateofappointment>='2020-01-01'
9 AND dateofappointment<='2020-12-31'
10 )
11 AS abc
12 JOIN
13 (
14 SELECT * FROM laboratory
15 ) AS l
16 ON l.labid=abc.labid GROUP BY labname
17 ORDER BY count(*)
18 LIMIT 3

```

Data Output Explain Messages Notifications

	labname character varying (20)	maxnumberofbookings bigint
1	Meditech Lab	1
2	Shreeji Lab	1
3	SRL DIAGNOSTICS	1

18. List the top 2 doctors whose patients have booked labs the minimum number of times.

### RELATIONAL ALGEBRA:

$$r1 \leftarrow \rho(a, (\text{appointment})) \bowtie_{\langle a.doctorid = d.doctorid \rangle} \rho(d, \text{doctor})$$

$$\bowtie_{\langle pr.patientId = a.patientId \text{ AND } pr.appointmentId = a.appointmentId \rangle}$$

$$\rho(pr, \sigma_{\text{pharmacyid is not null}}(\text{patientrecord}))$$

$$r2 \leftarrow \rho(d.doctorid, d.doctorId, d.firstName, COUNT(*) \rightarrow cnt)(r1)$$

$$r3 \leftarrow \mathcal{F}_{MIN(cnt)} \rightarrow mincnt(r2)$$

$$r4 \leftarrow \sigma_{cnt=mincnt}(r3 \times r4)$$

$$\text{result} \leftarrow \Pi_{\text{firstname}}(r4)$$

### SQL QUERY:

```

SELECT concat(firstName,' ',lastName)
FROM
(
    SELECT d.doctorId, d.firstName,d.lastName, COUNT(*) AS cnt
        FROM appointment AS a
        JOIN doctor AS d
        ON a.doctorId = d.doctorId
        JOIN patientRecord AS pr
        ON pr.patientId = a.patientId AND pr.appointmentId =
a.appointmentId
        WHERE pr.pharmacyId IS NOT NULL
        GROUP BY d.doctorId
        ORDER BY d.doctorId
    ) AS docCount
CROSS JOIN
(
    SELECT MIN(cnt) AS mincnt
    FROM
    (
        SELECT COUNT(*) AS cnt
        FROM appointment AS a
        JOIN doctor AS d
        ON a.doctorId = d.doctorId
        JOIN patientRecord AS pr
        ON pr.patientId = a.patientId AND pr.appointmentId =
a.appointmentId
        WHERE pr.pharmacyId IS NOT NULL
        GROUP BY d.doctorId
    ) AS docLabs
    )minNoOfLabs
WHERE docCount.cnt = minNoOfLabs.mincnt
ORDER BY docCount.doctorId
LIMIT 2

```



```

217 SELECT concat(firstName,' ',lastName) FROM
218 (
219 SELECT d.doctorId, d.firstName,d.lastName, COUNT(*) AS cnt
220 FROM appointment AS a
221 JOIN doctor AS d ON a.doctorId = d.doctorId
222 JOIN patientRecord AS pr ON pr.patientId = a.patientId AND pr.appointmentId =a.appointmentId
223 WHERE pr.pharmacyId IS NOT NULL
224 GROUP BY d.doctorId
225 ORDER BY d.doctorId
226 ) AS docCount CROSS JOIN
227 (SELECT MIN(cnt) AS mincnt FROM
228 (
229 SELECT COUNT(*) AS cnt
230 FROM appointment AS a
231 JOIN doctor AS d ON a.doctorId = d.doctorId
232 JOIN patientRecord AS pr ON pr.patientId = a.patientId AND pr.appointmentId =a.appointmentId
233 WHERE pr.pharmacyId IS NOT NULL
234 GROUP BY d.doctorId
235 ) AS docLabs) minNoOfLabs
236 WHERE docCount.cnt = minNoOfLabs.mincnt
237 ORDER BY docCount.doctorId
238 LIMIT 2
239

```

Data Output Explain Messages Notifications

	concat	
	text	
1	Naresh Trehan	
2	Suresh Advani	

19. List the average consultation fees a patient has to pay for a medical branch 'J'.

### RELATIONAL ALGEBRA:

$$r1 \leftarrow \Pi_{\text{medicalbranchid}}(\sigma_{\text{medicalbranchname}='Oncology'}(\text{medicalbranch}))$$

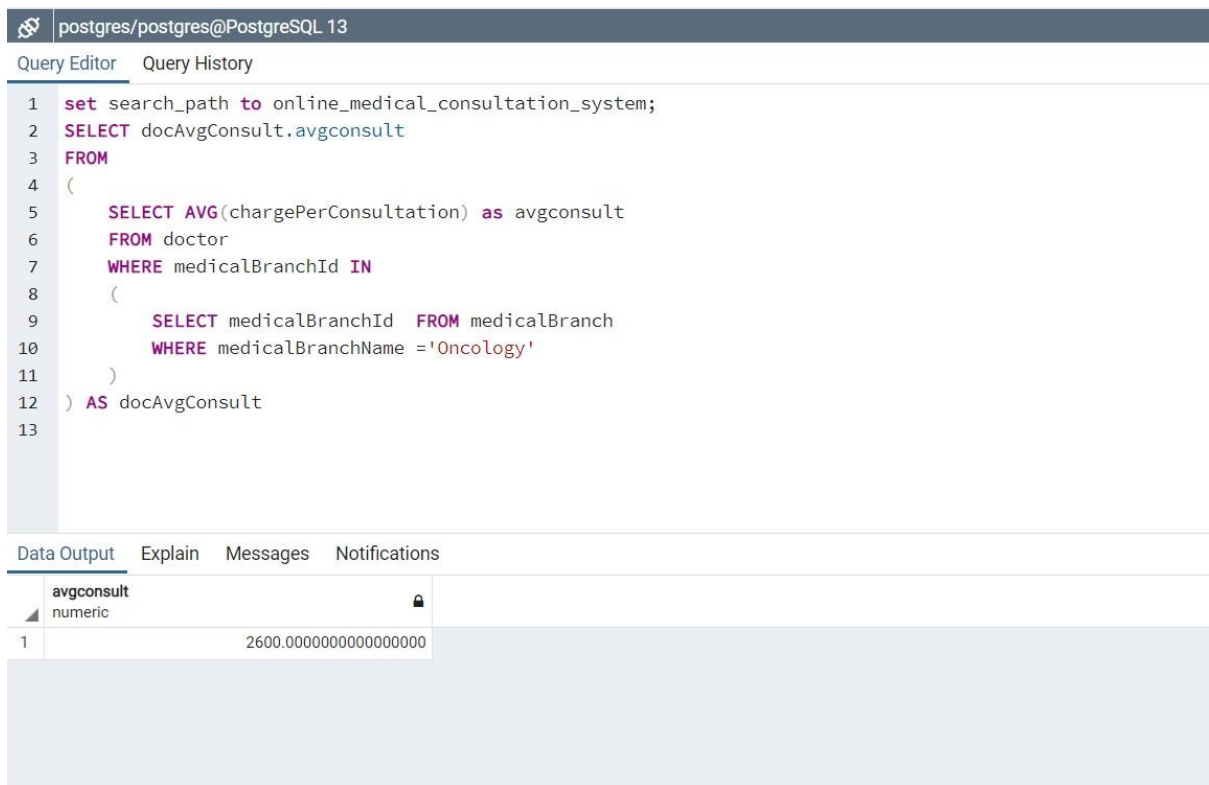
$$r2 \leftarrow \text{doctor SEMI-INTERSECTION}_{\text{doctor.medicalbranchid}=\text{medicalbranch.medicalbranchid}}(r1)$$

$$r3 \leftarrow \mathcal{F}_{\text{AVG}}(\text{chargePerConsultation}) \rightarrow \text{avgconsult}(r2)$$

$$\text{result} \leftarrow \Pi_{\text{avgconsult}}(r3)$$

## SQL QUERY:

```
SELECT docAvgConsult.avgconsult
FROM
(
    SELECT AVG(chargePerConsultation) as avgconsult
    FROM doctor
    WHERE medicalBranchId IN
    (
        SELECT medicalBranchId FROM medicalBranch
        WHERE medicalBranchName ='Oncology'
    )
) AS docAvgConsult
```



The screenshot shows a PostgreSQL Query Editor interface. The top bar indicates the connection is to 'postgres/postgres@PostgreSQL 13'. Below the bar, there are tabs for 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying a SQL query. The query is as follows:

```
1 set search_path to online_medical_consultation_system;
2 SELECT docAvgConsult.avgconsult
3 FROM
4 (
5     SELECT AVG(chargePerConsultation) as avgconsult
6     FROM doctor
7     WHERE medicalBranchId IN
8     (
9         SELECT medicalBranchId FROM medicalBranch
10        WHERE medicalBranchName ='Oncology'
11    )
12 ) AS docAvgConsult
13
```

Below the query editor, there are tabs for 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with one column, 'avgconsult', and one row, showing the value '2600.00000000000000000000'.

	avgconsult numeric
1	2600.00000000000000000000

20. Find the names of doctors who have sought online medical consultation as a patient also.

RELATIONAL ALGEBRA:

$\Pi_{\text{name} \rightarrow \text{common\_name}}(\text{doctor}) \cap \Pi_{\text{name} \rightarrow \text{common\_name}}(\text{patient})$

SQL QUERY:

```
SELECT firstName || ' ' || lastName as common_name FROM doctor
INTERSECT
SELECT firstName || ' ' || lastName as common_name FROM patient;
```



The screenshot shows a PostgreSQL query editor interface. The title bar indicates the connection is to 'postgres/postgres@PostgreSQL 13'. The 'Query Editor' tab is active, displaying the following SQL query:

```
1 SET SEARCH_PATH TO online_medical_consultation_system;
2 SELECT firstName || ' ' || lastName as common_name FROM doctor
3 INTERSECT
4 SELECT firstName || ' ' || lastName as common_name FROM patient;
5
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with the following structure:

	common_name
1	Janu Shah

21. For sending promotional mails by the hospital, retrieve a list of email ids and mobile numbers of all patients and doctors.

### RELATIONAL ALGEBRA:

$\Pi_{\text{emailid, mobilenumbr}}(\text{patient})$

$\cup$

$\Pi_{\text{emailid, mobilenumbr}}(\text{doctor})$

### SQL QUERY:

SELECT emailid, mobilenumbr from patient

UNION



SELECT emailid, mobilenumbr from doctor;

postgres/postgres@PostgreSQL 13

Query EditorQuery History

```
1 SET SEARCH_PATH TO online_medical_consultation_system;
2 SELECT emailid, mobilenumbr from patient
3 UNION
4 SELECT emailid, mobilenumbr from doctor;
5
```

Data OutputExplainMessagesNotifications

	 emailid character varying (25)	 mobilenumbr numeric (10)	
1	doctor02@gmail.com	9815479515	
2	doctor08@gmail.com	8745684128	
3	hc24@gmail.com	7654321098	
4	doctor04@gmail.com	9564871259	
5	doctor03@gmail.com	6854795648	
6	pn14@gmail.com	8849424642	
7	dg56@gmail.com	981234567	
8	doctor06@gmail.com	6875941258	
9	doctor07@gmail.com	9875123645	
10	ns12@gmail.com	4569870123	
11	sk18@gmail.com	3216549870	
12	doctor09@gmail.com	8745659728	
13	kp123@gmail.com	9876543210	
14	sm839@gmail.com	9702461510	

22. Retrieve all patient records and their pharmacy details.

### RELATIONAL ALGEBRA:

$r1 \leftarrow \rho(\text{pr}, \text{patientrecord}) \bowtie_{\text{pr.pharmacyid}=\text{ph.pharmacyid}} \rho(\text{ph}, \text{pharmacy})$   
 $\text{result} \leftarrow \Pi(r1)$

### SQL QUERY:

SELECT \* FROM patientrecord AS pr  
 LEFT JOIN pharmacy AS ph ON(pr.pharmacyid=ph.pharmacyid);

143	Set search_path to online_medical_consultation_system
144	
145	
146	SELECT * FROM patientrecord AS pr
147	LEFT JOIN pharmacy AS ph ON(pr.pharmacyid=ph.pharmacyid);
148	

	patientid character varying (10)	appointmentid character varying (10)	doctorobservation character varying (100)	doctordiagnosis character varying (100)	orderedmedicine boolean	labreport character varying (50)	digitalprescription character varying (50)
1	P01	A01			true	report.pdf	photo.jpg
2	P02	A02	follow the medicine routine	viral fever	true	report1.pdf	photo1.jpg
3	P03	A03	take appointment after one w...	skin disease	false		photo3.jpg
4	P04	A04			true	report3.pdf	
5	P05	A05	get admit in hospital	malaria	false	report4.pdf	photo4.jpg
6	P06	A06	get admit	dengue	true	report5.pdf	photo5.jpg
7	P07	A07	normal pain in head due to m...	headache	true		photo7.jpg
8	P08	A08	get admit in hospital	typhoid	true	report8.pdf	
9	P09	A09	take the given medicine	viral cold	false		photo9.jpg
10	P10	A10	due to lack of protien	weakness	true		photo10.jpg
11	P10	A11	take the given medicine	weakness	false		photo10.jpg
12	P08	A12	due to lack of protien	viral cold	true		photo10.jpg
13	P11	A11	get admit	dengue	false	report11.pdf	photo11.jpg
14	P12	A12	normal pain in head due to m...	headache	true		photo12.jpg
15	P13	A113	get admit in hospital	typhoid	false	report13.pdf	

23. Retrieve Patient Ids who have completed appointments without booking pharmacy.

### RELATIONAL ALGEBRA:

$r1 \leftarrow (\rho(\text{pr}, \text{patientRecord})) \bowtie_{\langle \text{pr.pharmacyId} = \text{ph.pharmacyId} \rangle} \rho(\text{ph}, \text{pharmacy})$

$r2 \leftarrow \text{Appointment SEMI-DIFFERENCE}_{\langle \text{patientId} = \text{patientId} \rangle} r1$

$\text{Result} \leftarrow \Pi_{\text{firstName}} (r2 \bowtie_{\langle r2.\text{patientId} = p.\text{patientId} \rangle} \rho(p, \text{patient}))$

### SQL QUERY:

```
SELECT p.firstName
FROM patient AS p
JOIN
(
    SELECT DISTINCT patientId
    FROM appointment
    WHERE patientId NOT IN
        (SELECT DISTINCT pr.patientId
         FROM patientRecord AS pr
         JOIN pharmacy AS ph
         ON pr.pharmacyId = ph.pharmacyId)
) AS app
ON p.patientId = app.patientId;
```

postgres/postgres@PostgreSQL 13

Query Editor   Query History

```

1 SET SEARCH_PATH TO online_medical_consultation_system;
2
3 SELECT p.firstName
4 FROM patient AS p
5 JOIN
6 (
7     SELECT DISTINCT patientId
8     FROM appointment
9     WHERE patientId NOT IN
10         (SELECT DISTINCT pr.patientId
11          FROM patientRecord AS pr
12          JOIN pharmacy AS ph
13           ON pr.pharmacyId = ph.pharmacyId)
14 ) AS app
15 ON p.patientId = app.patientId
16

```

Data Output   Explain   Messages   Notifications

	firstname	
	character varying (20)	
1	Shubham	
2	Swati	

24. List doctor ids of doctors who have conducted all appointments for the medical branch='MB04'.

### RELATIONAL ALGEBRA:

$r_0 \leftarrow \text{appointment} \times (\rho(a_1, \sigma_{\text{medicalbranchid}='MB04'}(\text{doctor})))$   
 $r_1 \leftarrow \Pi_{a_1.\text{doctorid}, \text{appointmentid}}(r_0) - \Pi_{\text{doctorid}, \text{appointmentid}}(\text{appointment})$   
 $r_2 \leftarrow \Pi_{\text{doctorid}}(r_1)$   
 $\text{result} \leftarrow \Pi_{\text{doctorid}}(\text{appointment SEMI-JOIN}_{\text{appointment.doctorid}=r_2.\text{doctorid}} r_2)$

### SQL QUERY:

select distinct(doctorid) from appointment

where doctorid in

(

```

select doctorid from
(
    select a1.doctorid, appointmentid from appointment
    cross join (select * from doctor where medicalbranchid='MB04') as
a1
    except
    select doctorid, appointmentid from appointment
) as a2
);

```

Query Editor

Query History

```

173 Set search_path to online_medical_consultation_system
174
175 select distinct(doctorid) from appointment
176 where doctorid in
177 (
178     select doctorid from
179     (
180         select a1.doctorid, appointmentid from appointment
181         cross join (select * from doctor where medicalbranchid='MB04') as a1
182         except
183         select doctorid, appointmentid from appointment
184     ) as a2
185 );
186

```

Data Output

Explain

Messages

Notifications

	doctorid character varying (5)	
1	D11	
2	D12	
3	D05	
4	D04	

25. Retrieve details of such patients whose firstname begin with 'J' and lastname begins with 'S'.

### RELATIONAL ALGEBRA:

$\sigma_{\text{firstname LIKE}(J\%) \text{ AND lastname LIKE}(S\%)}(\text{patient})$

### SQL QUERY:



```
select * from patient where firstname like 'J%' and lastname like 'S%';
```

```
189
190 Set search_path to online_medical_consultation_system
191
192
193 select * from patient where firstname like 'j%' and lastname like 's%';
194
195
```

[Data Output](#) [Explain](#) [Messages](#) [Notifications](#)

[illegible]