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:

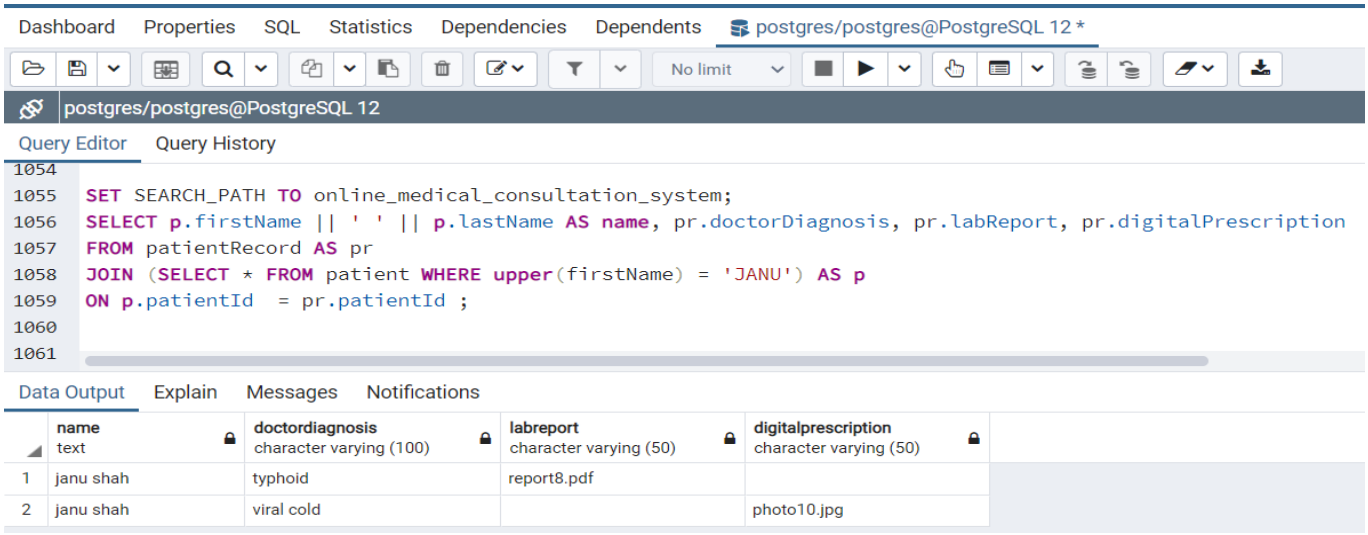
1. For a patient ‘XYZ’, retrieve doctor’s diagnosis, digital prescription and lab reports.

RELATIONAL ALGEBRA:

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

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 ;
```



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

RELATIONAL ALGEBRA:

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

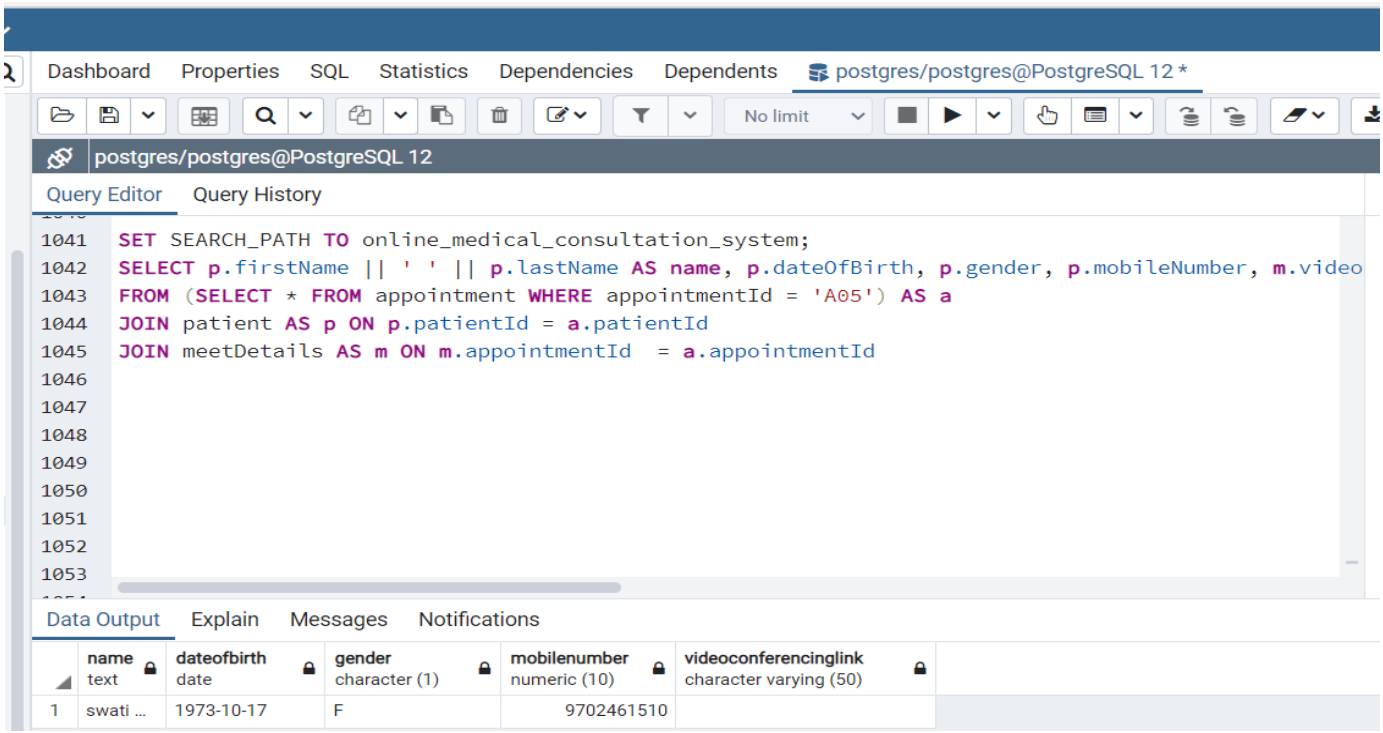
SQL QUERY:

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



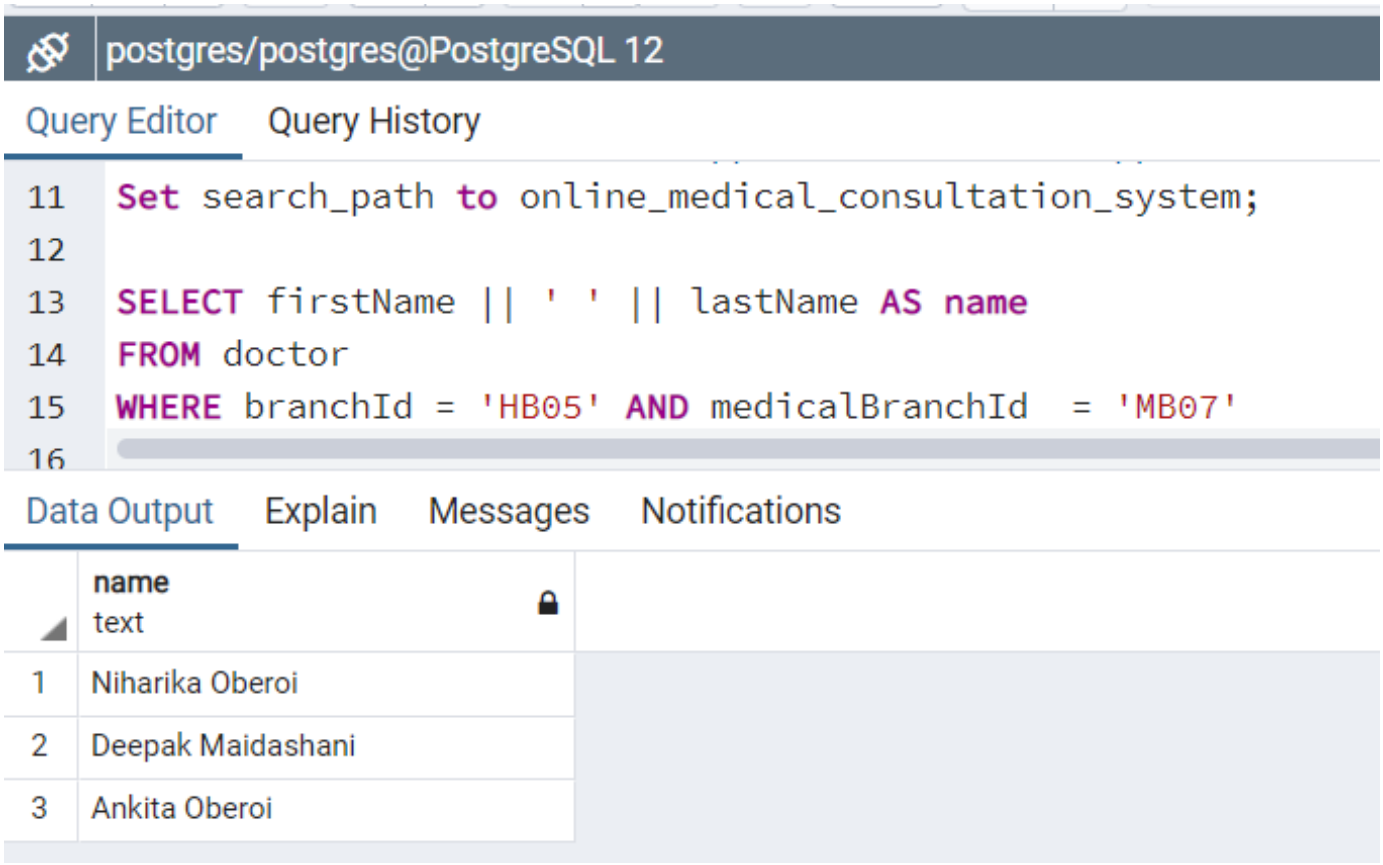
3. List all doctors from ‘MB07’ medical branch working at ‘HB05’ branch of the hospital.

RELATIONAL ALGEBRA:

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

SQL QUERY:

```
SELECT firstName || ' ' || lastName AS name
FROM doctor
WHERE branchId = 'HB05' AND medicalBranchId  = 'MB07'
```



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

RELATIONAL ALGEBRA:

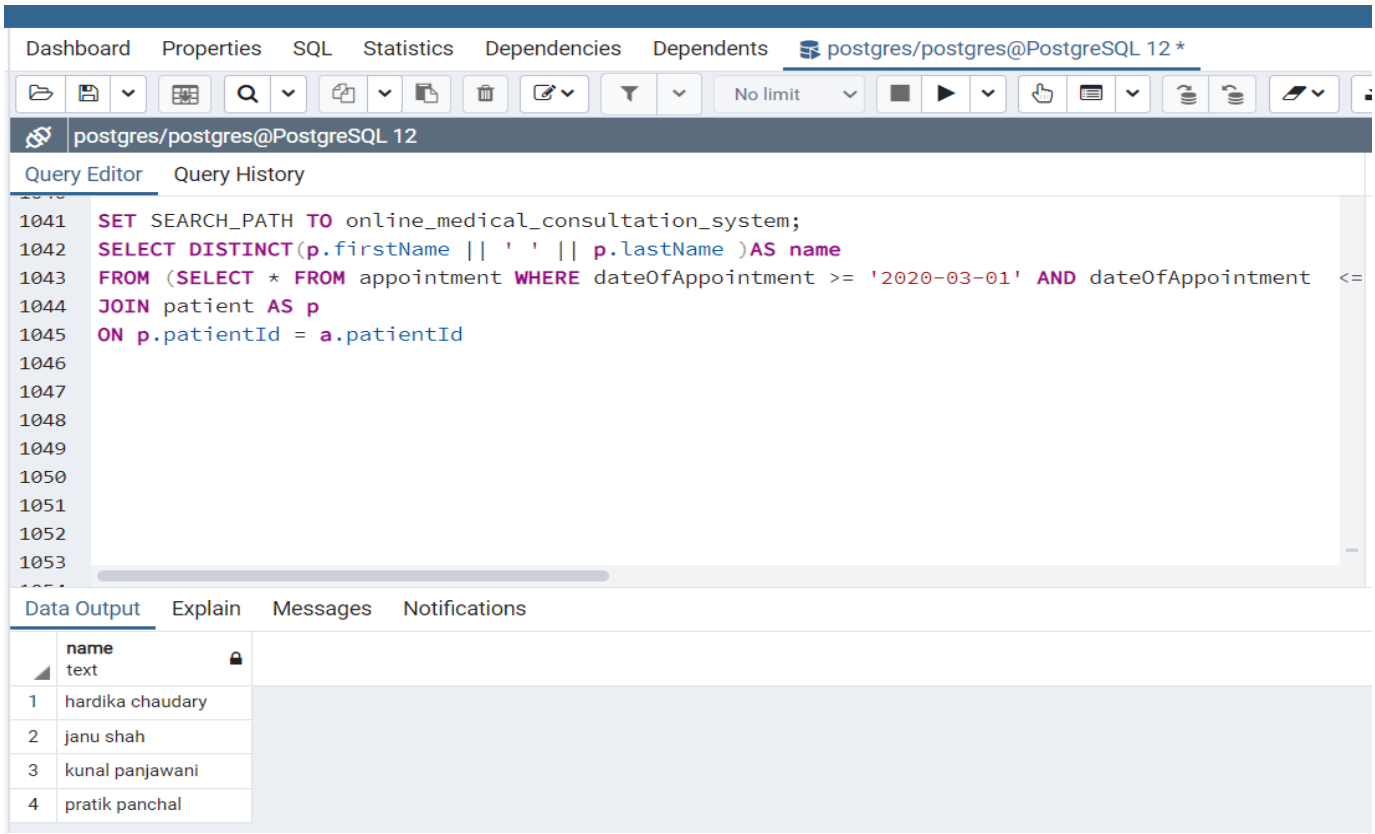
Π_{name} ((ρ (a, $\sigma_{dateOfAppointment >= '2020\text{-}03\text{-}01' \text{ AND } dateOfAppointment <= CURRENT\_DATE}$ (appointment))) $\bowtie_{<a.patientId=p.patientId>}$ ρ (p, patient))

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



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

<u>RELATIONAL ALGEBRA:</u>

$\mathcal{F}_{\text{COUNT(*)-> numberOfConsultations}}$ ((ρ (a, $\sigma_{\text{dateOfAppointment >= '2018-01-01' AND dateOfAppointment <= '2020-12-31'}}$ (appointment))) ⋈<a.patientId=p.patientId> ρ (p, $\sigma_{\text{firstName='KUNAL'}}$patient))

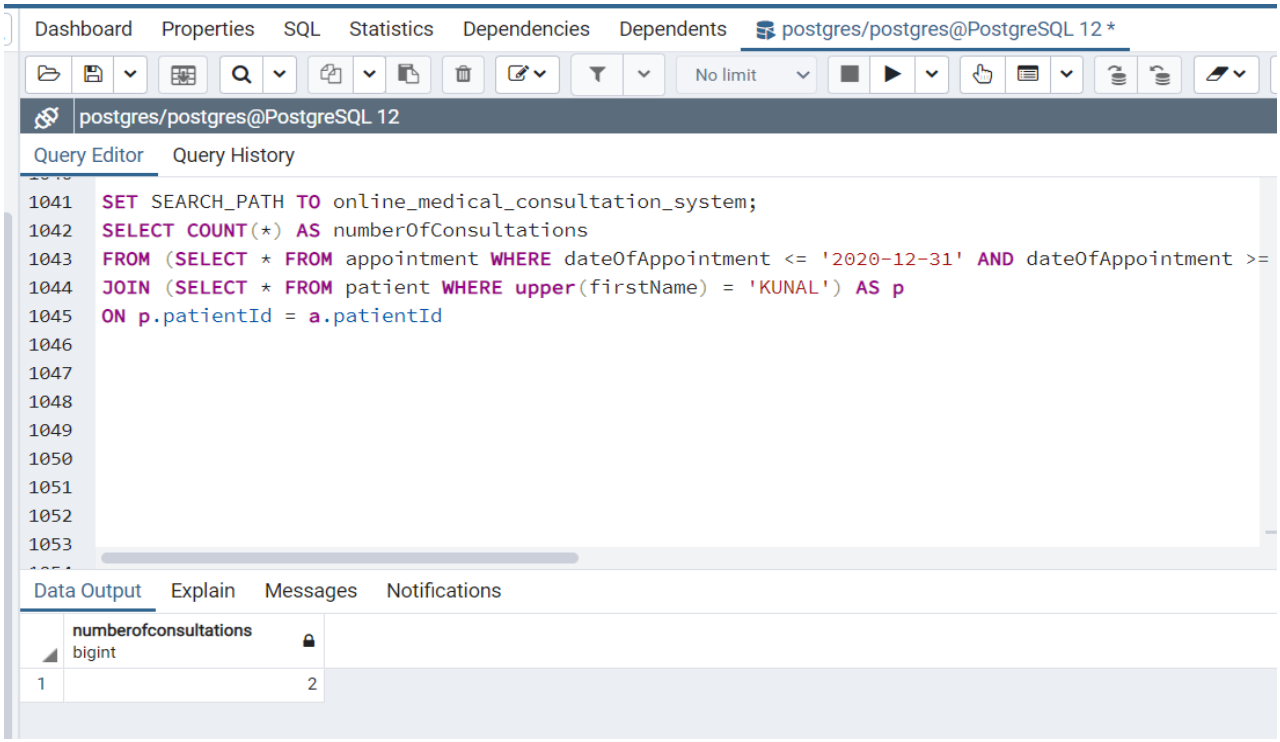
<u>SQL QUERY:</u>

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



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

<u>RELATIONAL ALGEBRA:</u>

r1<- ρ (a, $\sigma_{\text{dateOfAppointment >= ’2020-01-01’ AND dateOfAppointment <= ‘2020-12-31’}}$ (appointment))

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

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

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

r3<- $\mathcal{F}_{MAX(cnt) \rightarrow maxcnt}$ (r2)

r4<- $\sigma_{cnt= cnt}$ (r3 X r2)

result<- Π_{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

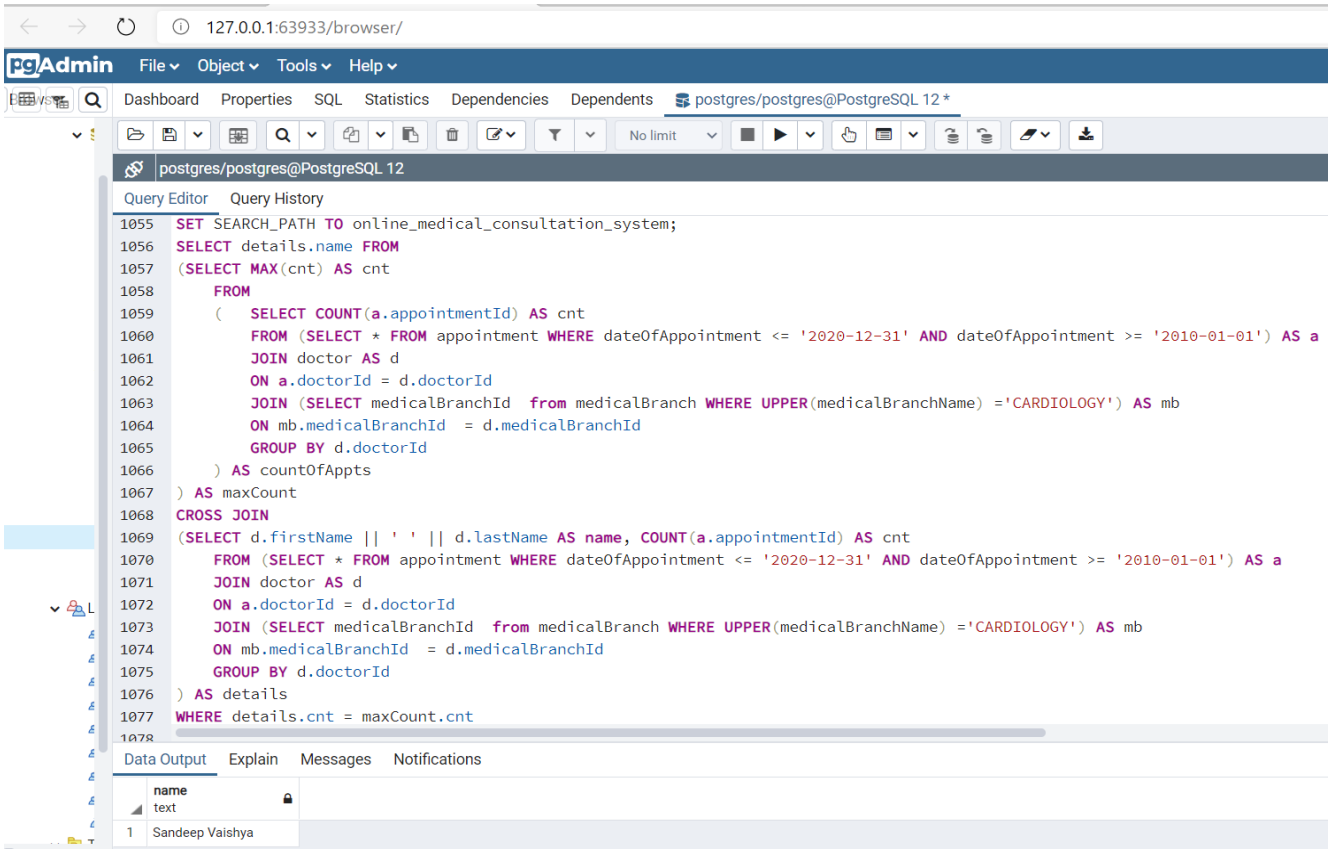
JOIN (SELECT medicalBranchId from medicalBranch WHERE UPPER(medicalBranchName) ='CARDIOLOGY') AS mb

ON mb.medicalBranchId = d.medicalBranchId

GROUP BY d.doctorId

) AS details

WHERE details.cnt = maxCount.cnt



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

RELATIONAL ALGEBRA:

$r1 \leftarrow \rho(d, doctor)$

$\bowtie_{<d.branchId = h.branchId>} \rho(h, hospitalBranch)$

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

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

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

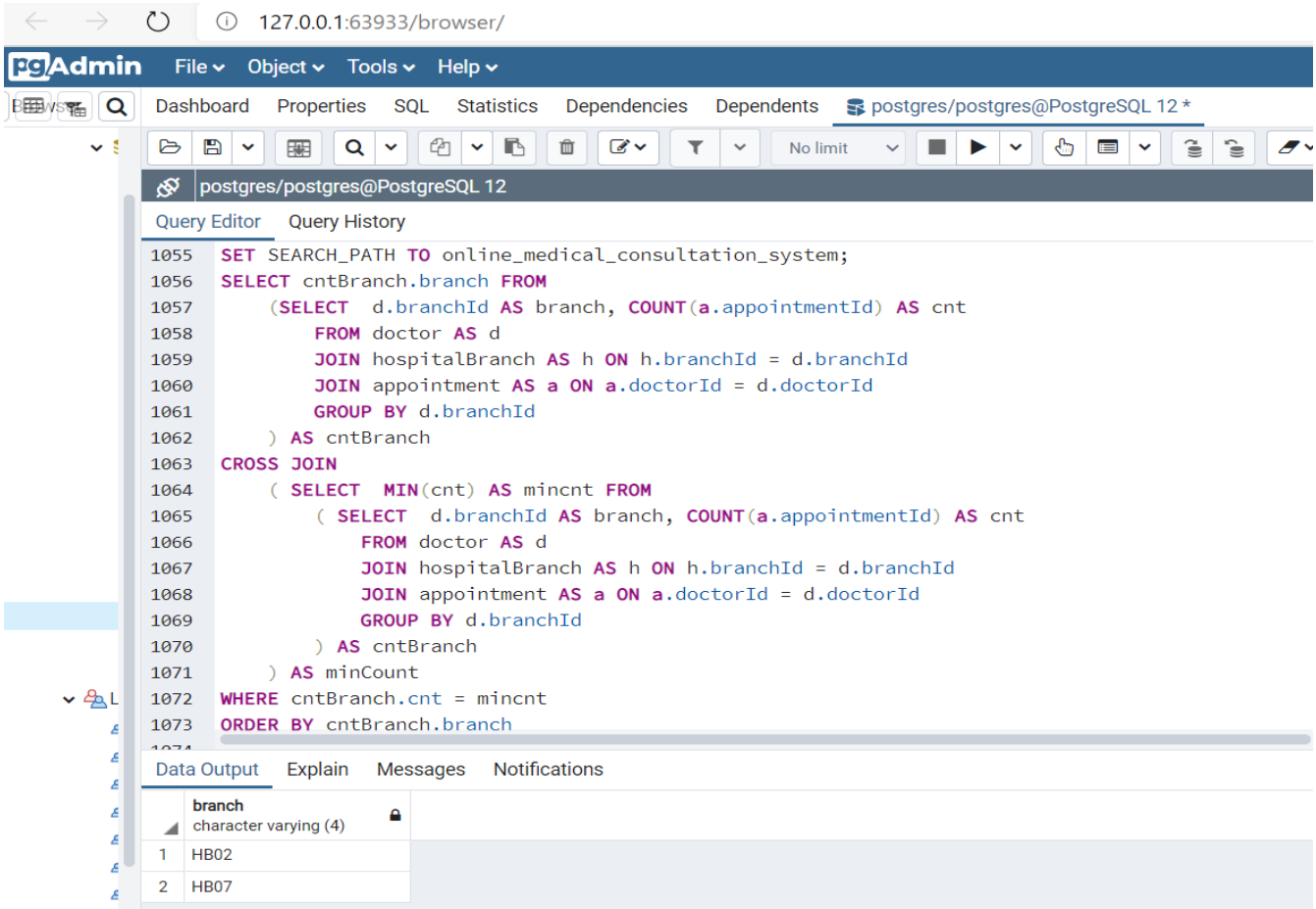
$r4 \leftarrow \sigma_{cnt = mincnt}(r3 \times r2)$

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

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
```



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

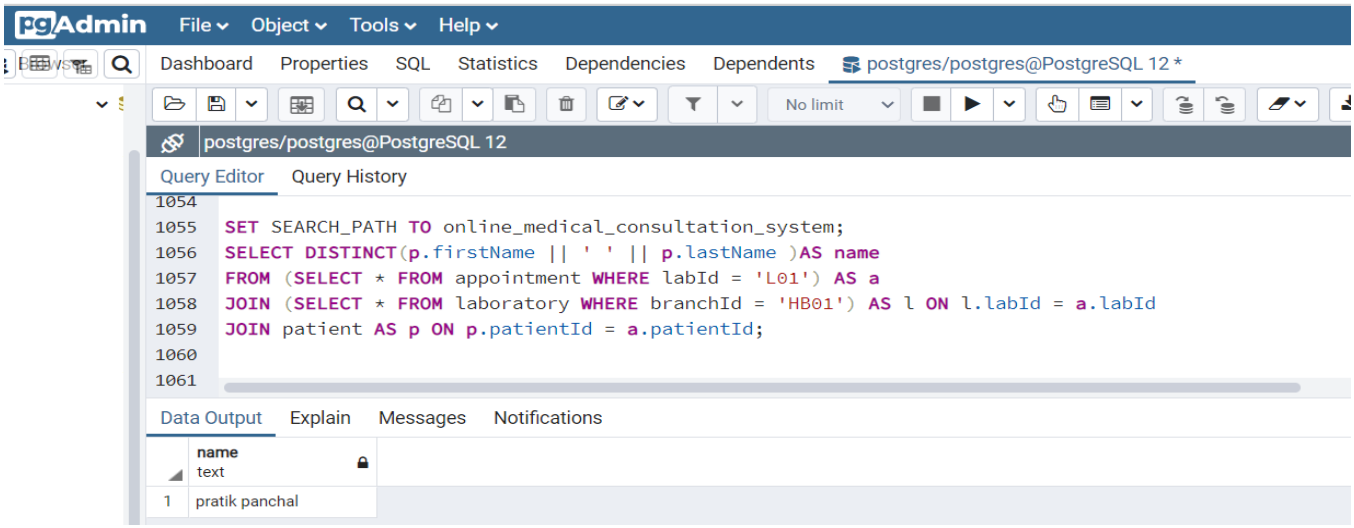
RELATIONAL ALGEBRA:

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

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

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
```



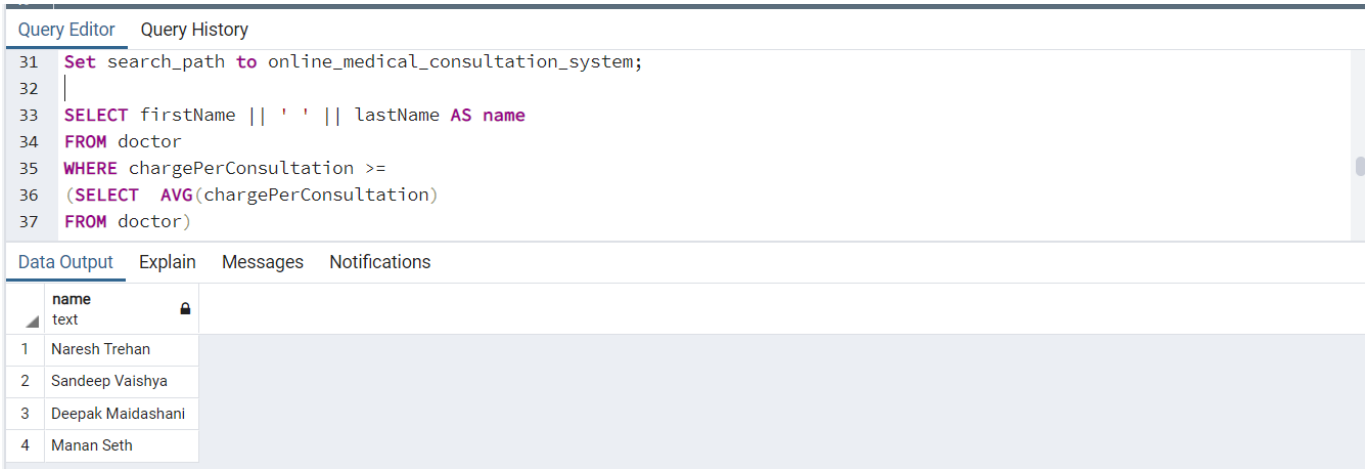
9. List the doctors with consultation charges more than the average consultation charges.

RELATIONAL ALGEBRA:

$\Pi_{name}\ \sigma_{chargePerConsultation>=(\mathcal{F}AVG(chargePerConsultation)\ (doctor))}\ (doctor)$

SQL QUERY:

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



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

RELATIONAL ALGEBRA:

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

$\bowtie_{<a.appointmentId=p.\ appointmentId>}$ ρ (py, payment)

$\bowtie_{<a.patientId=p.patientId>}$ ρ (p, patient)

r2<- $_{a.patientId}\mathcal{F}_{a.patientId}$ (r1)

r3<- $\sigma_{SUM(py.transactionAmount)>5000}$(r2)

r4<-$\Pi_{patientId}$(r3)

result<- patient SEMI-INTERSECTION $_{<patientId = patientId>}$(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:

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

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

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

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;
```

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

RELATIONAL ALGEBRA:

$r1 \leftarrow \sigma$(patientrecord $\bowtie_{<patientrecord.patientid=pharmacy.pharmacyid>}$ pharmacy)

result$\leftarrow \Pi_{patient.name\ ,\ patient.lastname,\ patient.dateOfbirth,\ patient.gender,\ patient.mobileNumber}$ (patient $\bowtie_{<patient.patientid=r1.patientid>}$ 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:

$r1 \leftarrow \rho(a, appointment)$

$\bowtie_{<a.patientId = p.patientId>} \rho(p, \sigma_{patientname='kunal'}(patient))$

$\bowtie_{<a.doctorId = d.doctorId>} \rho(d, \sigma_{firstname='Janu'}(doctor))$

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

r3<- feedback SEMI-INTERSECTION$_{feedback.appointmentid=a.appointmentid}$(r2)
result<- $\Pi_{reviewbypatient}$(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:

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

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

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

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

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

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

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.

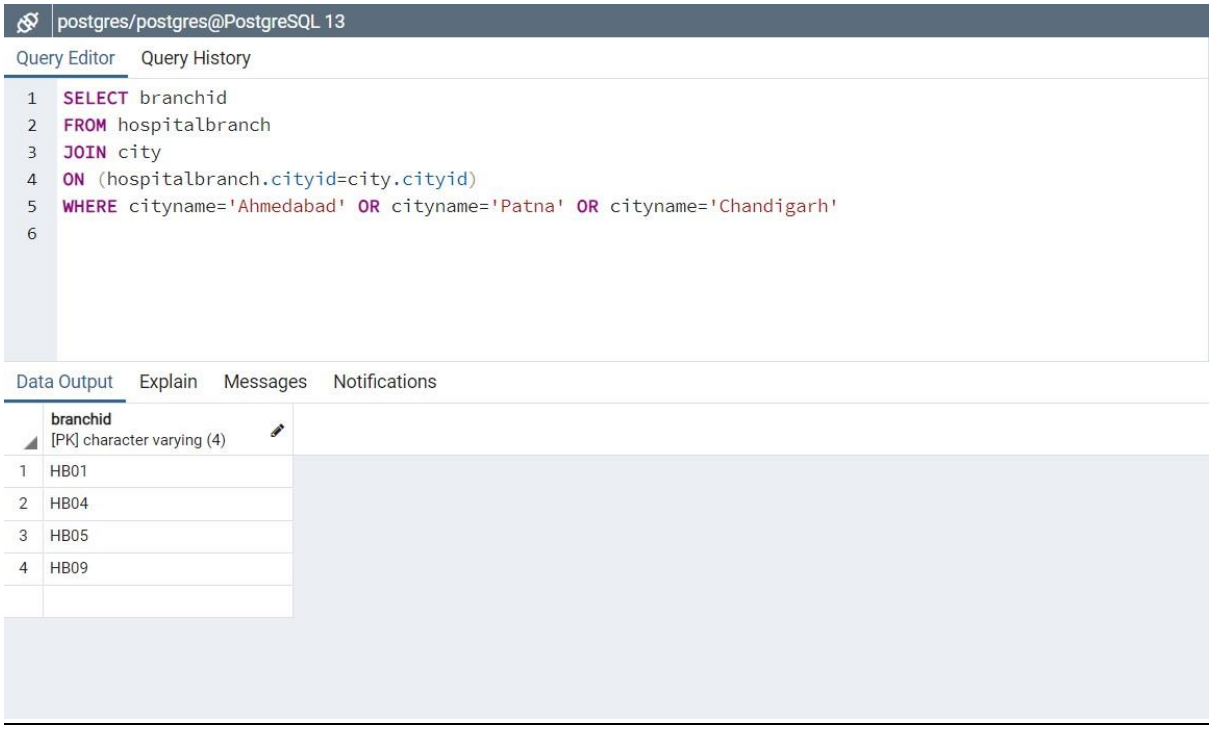
RELATIONAL ALGEBRA:

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

SQL QUERY:

SELECT branchid
FROM hospitalbranch

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



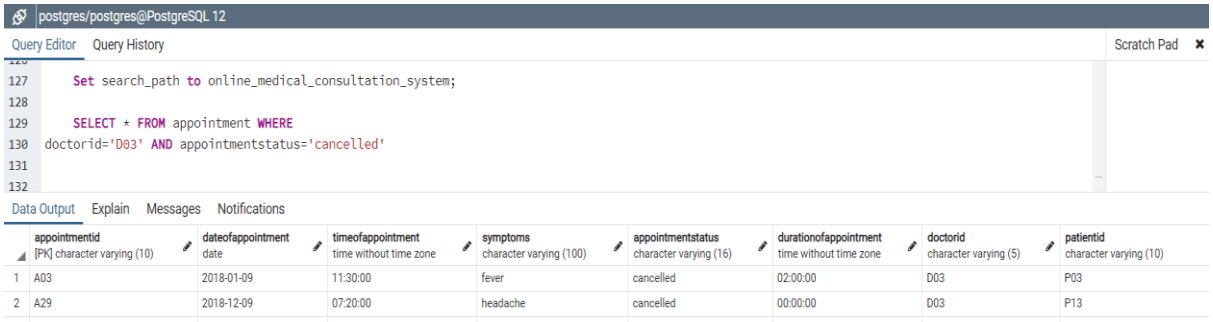
16. List the appointments having appointment status as ‘cancelled’ for a doctor ‘A’.

RELATIONAL ALGEBRA:

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

SQL QUERY:

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



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

((ρ (a, σ$_{\text{dateOfAppointment}}$ >= ’2020-01-01’

AND dateOfAppointment <= ‘2020-12-31’ (appointment)))

⋈$_{\text{<abc.labid=l.labId>}}$$\mathcal{F}$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:

$r1 \leftarrow \rho(a,(appointment)) \bowtie_{<a.doctorid=d.doctorid>} \rho(d,doctor)$

$\bowtie_{<pr.patientId = a.patientId \text{ AND } pr.appointmentId = a.appointmentId>}$

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

$r2 \leftarrow {}_{d.doctorid}\mathcal{F}_{d.doctorId,\ d.firstName,COUNT(*)->cnt}(r1)$

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

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

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

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
```

Query Editor | Query History

```
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
```

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<- $\Pi_{\text{medicalbranchid}}(\sigma_{\text{medicalbranchname='Oncology'}}(\text{medicalbranch}))$

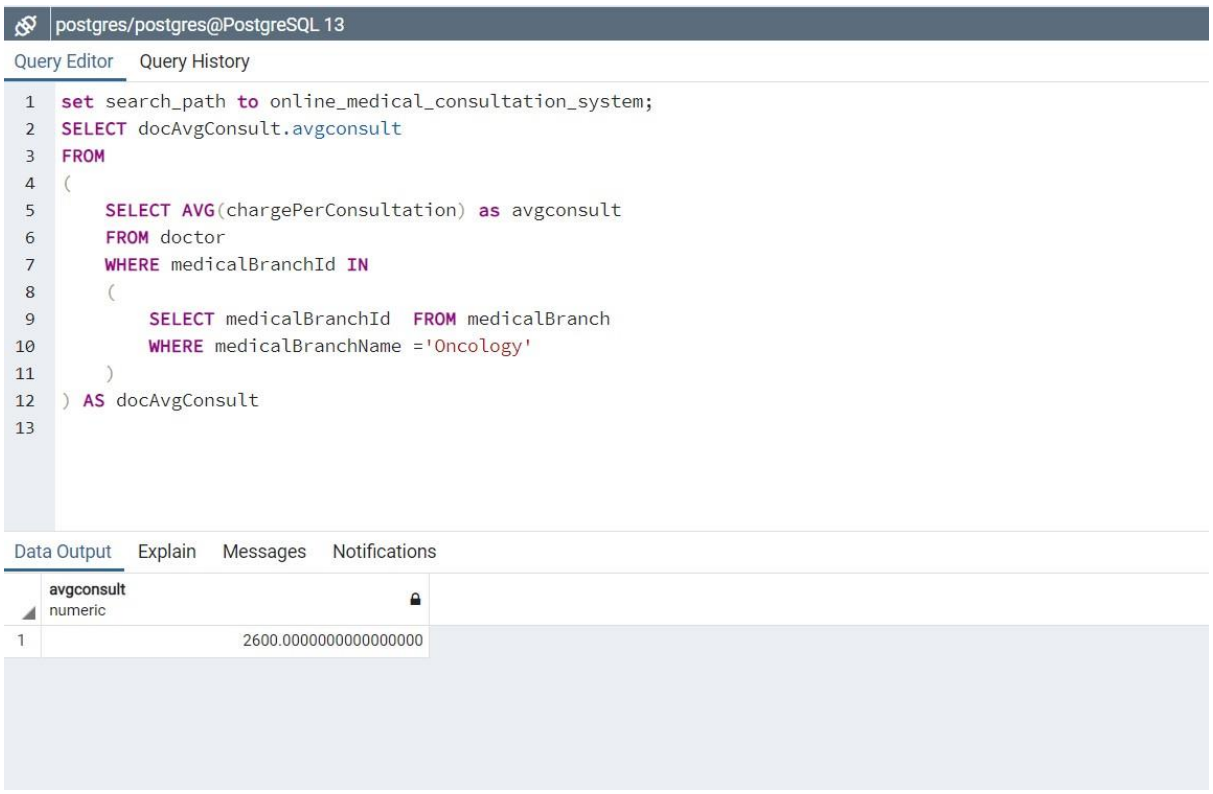
r2<- doctor SEMI-INTERSECTION$_{\text{doctor.medicalbranchid=medicalbranch.medicalbranchid}}$(r1)

r3<- $\mathcal{F}_{\text{AVG(chargePerConsultation) -> avgconsult}}$ (r2)

result<- $\Pi_{\text{avgconsult}}$ (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
```



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

RELATIONAL ALGEBRA:

$\Pi_{\text{name -> common\_name}}(\texttt{doctor}) \cap \Pi_{\text{name -> common\_name}}(\text{patient})$

SQL QUERY:

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

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

RELATIONAL ALGEBRA:

$\Pi_{emailid,\ mobilenumber}(patient)$

$\cup$

$\Pi_{emailid,\ mobilenumber}(doctor)$

SQL QUERY:

```
SELECT emailid, mobilenumber from patient
UNION
SELECT emailid, mobilenumber from doctor;
```

postgres/postgres@PostgreSQL 13

Query Editor | Query History

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

Data Output | Explain | Messages | Notifications

	emailid character varying (25)	mobilenumber 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:

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

$result \leftarrow \Pi(r1)$

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
```

Data Output | Explain | Messages | Notifications

	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	maleria	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:

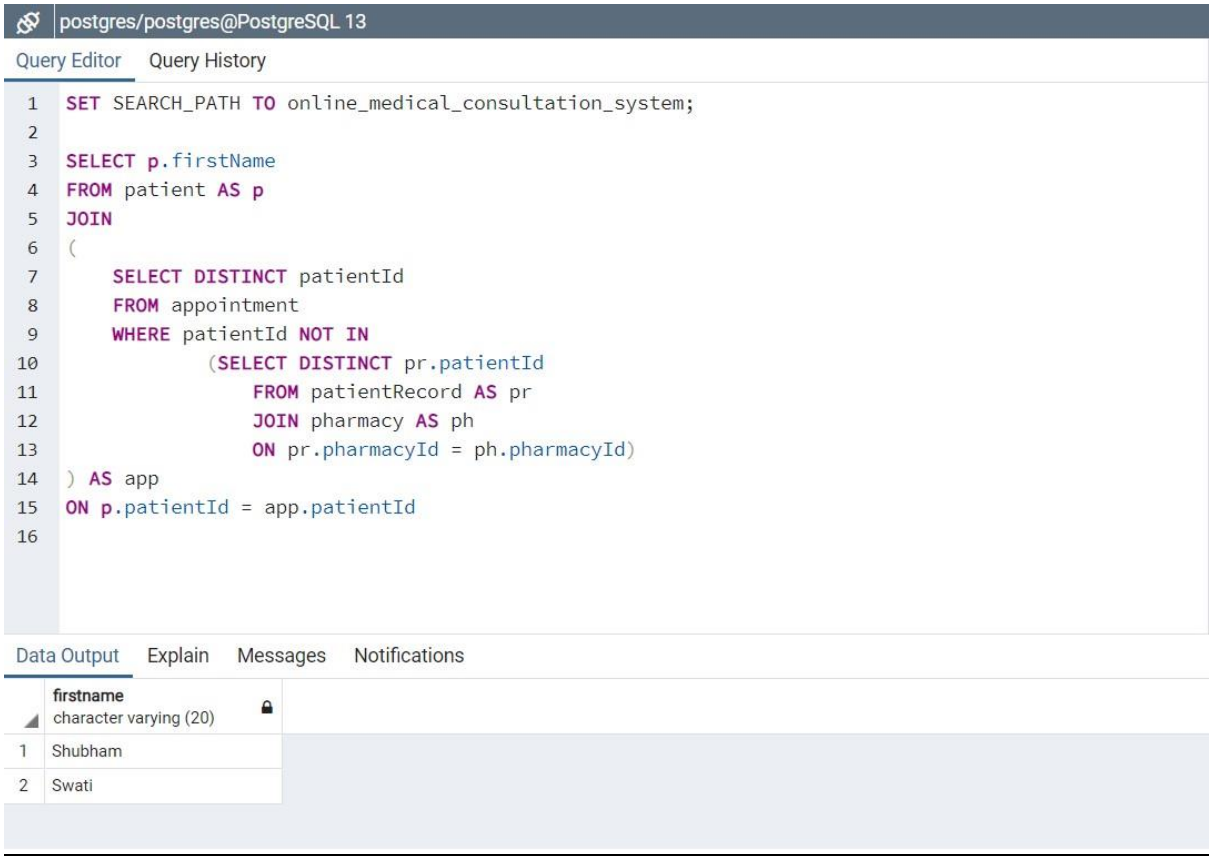
r1<- (ρ (pr, patientRecord))$\bowtie_{<pr.pharmacyId=ph.pharmacyId>}$ρ(ph, pharmacy))

r2<- Appointment SEMI-DIFFERENCE$_{<patientId=patientId>}$r1

Result<- $\Pi_{firstName}$ (r2 $\bowtie_{<r2.patientId =p.patientId>}$ ρ (p,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;
```



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

RELATIONAL ALGEBRA:

r0<- appointment X ($\rho(a1,\sigma_{medicalbranchid='MB04'}(doctor))$)
r1<- $\Pi_{a1.doctorid,appointmentid}(r0) - \Pi_{doctorid,appointmentid}(appointment)$
r2<- $\Pi_{doctorid}(r1)$
result<-$\Pi_{doctorid}(appointment\ SEMI\text{-}JOIN_{appointment.doctorid=r2.doctorid}\ r2)$

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
);
```

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

RELATIONAL ALGEBRA:

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

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

	patientid [PK] character varying (10)	firstname character varying (20)	lastname character varying (20)	dateofbirth date	gender character (1)	mobilenumber numeric (10)	address character varying (100)	emailid character varying (25)	passw charac
1	P08	janu	shah	2000-08-23	F	1236547890	403, Ivory Terrace, Alkapuri,va...	janu@gmail.com	js@43
2	P21	jayanti	sharma	1993-01-17	M	9791273710	c-24/a Ragvilas Soc, B-24/a R...	mm839@gmail.com	mmeh