Hospital Information System

I have read and understood the course academic integrity policy in the syllabus of the class.

1. Team details

Name - Shruti Gupta
Ubname - sgupta55
Ubmail - sgupta55@buffalo.edu

Name - Pavana Lakshmi Venugopal
Ubname - pavanala
Ubmail - pavanala@buffalo.edu

Name - Rounak Biswas
Ubname - rounakbi
Ubmail - rounakbi@buffalo.edu

2. Problem Statement

For the hospital to handle patient information, medical records, supplies, and other routine data, a database management system must be effective and dependable. The manual nature of the present system makes it prone to mistakes, which causes delays in patient treatment, poor departmental collaboration, and inefficient use of resources. The administration of the hospital requires a system that can handle procedures, safely keep data, give users access to information in real-time, and produce reports for analysis and decision-making. The system should also be simple to use and adaptable to the unique requirements and processes of the facility. The hospital is looking for a solution that will increase efficiency all around, better patient treatment, and streamline processes.

3. Current Relational Schemas

1. Relation Schema: patient(p_id, first_name, last_name, gender, phone_number, dob,emergency_contact, address, email)
Primary key: p_id

2. Relation Schema: employee(e_id, e_type, first_name, last_name, dob, address, gender, phone_number, emergency_contact, date_of_joining, date_of_resigning)
Primary key: e_id

3. Relation Schema: Doctors(e_id, dept_no)
Primary key: e_id
Foreign key: e_id from employee relation, dept_no from dept relation

4. Relation Schema: dept(dept_no, dept_name, no_of_doctors)
Primary key: Dept_no

5. Relation Schema: general_personal(e_id, charge/hr)
Primary key: e_id
Foreign key: e_id from employee relation

6. Relation Schema: opd_ Doctor(e_id, cons_fee)
Primary key: e_id
Foreign key: e_id from doctor relation

7. Relation Schema: duty_assigned(e_id, days, start_shift, end_shift)
Primary key: e_id, days
Foreign key: e_id from OPD_ Doctor relation

8. Relation Schema: duty_attendance(e_id, dates, swipe_in, swipe_out)
Primary key: e_id, dates
Foreign key: e_id from opd_doctor relation

9. Relation Schema: patient_medical_history(p_id, medical_history)
Primary key: p_id
Foreign key: p_id from patient relation

10. Relation Schema: room_detail(room_no, room_type, occupied_status, cost)
Primary key: room_no
Foreign key: room_type from room_type relation

11. Relation Schema: room_type(room_type, room_cost)
Primary key: room_type

12. Relation Schema: patient_admit(case_id_admit, p_id, admit_date, discharge_date, room_no)
Primary key: case_id_admit
Foreign key: p_id from patient relation; room_no from room_details relation

13. Relation Schema: policy(policy_no, policy_name, claim_amt)
Primary key: policy_no

14. Relation Schema: patient_insurance(policy_no, p_id)
Primary key: p_id, policy_no
Foreign key: policy_no from Policy relation; p_id from patient relation

15. Relation Schema: provided_healthcare_service(service_id, dept_no, service_desc, service_charge, service_code)
Primary key: service_id
Foreign key: dept_no from department relation

16. Relation Schema: discharge_summary(case_id_admit, p_id, admit_diagnosis)
Primary key: Case_id_admit
Foreign key: case_id_admit from Patient_admit relation; p_id from Patient relation

17. Relation Schema: opd_bill(case_id_opd, p_id, e_id, opd_charge, diagnosis)
Primary key: case_id_opd
Foreign key: p_id from patient relation; e_id from opd_doctor relation

18. Relation Schema: interim_bill(case_id_admit, bill_id, p_id, room_cost, service_charge, opd_charge)
Primary: bill_id, case_id_admit
Foreign key: p_id from patient relation, case_id_admit from patient_admit relation, case_id_opd from opd_bill

19. Relation Schema: total_bill(case_id_admit, p_id, policy_no, claim_accepted, total_charge, payable_amt)
Primary key: case_id_admit
Foreign key: case_id_admit from patient_admit relation; p_id from patient relation; policy_no from policy relation

4. Normalization

Minimal FD sets and BCNF

1) Patient relation

P_id→first_name
P_id→last_name
P_id→gender
P_id→phone_number
P_id→DOB
P_id→emergency_contact
P_id→address
P_id→email

Key={P_id}

The patient relation does not have multi-valued attributes hence it is in 1NF.
The patient relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.

The following three relations are MVDs and hence, violate 4NF.
P_id→Phone_number
P_id→emergency_contact

So
R1(P_id, Phone_number)
R2(P_id,emergency_contact)
R3(p_id, first_name, last_name, gender, dob, address, email)

2) Employee relation

E_id→first_name
E_id→last_name
E_id→DOB
E_id→address
E_id→gende
E_id→phone_number
E_id→emergency_contact
E_id→Date_of_joining
E_id→Date_of_resigning

key{E_id}

The employee relation does not have multi-valued attributes hence it is in 1NF.
The employee relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.

The following relations are MVDs and hence, violate 4NF.
E_id→Phone_number
E_id→emergency_number
So
R1(E_id, Phone_number)
R2(E_id, emergency_number)
R3(E_id,e_type, first_name, last_name, dob, address, gender, date_of_joining, date_of_resigning)

3) Doctors relation

E_id→dept_no

key{E_id}

The doctors relation does not have multi-valued attributes hence it is in 1NF.
The doctors relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(E_id, dept_no)

4) Dept relation

dept_no →Dept_Name
dept_no →no_of_doctors

Key{dept_no}

The dept relation does not have multi-valued attributes hence it is in 1NF.
The dept relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.

So
R(dept_no, Dept_name, no_of_doctors)

5) general_personal relation

E_id→Charge/hr

Key{E_id}

The general_personal relation does not have multi-valued attributes hence it is in 1NF.
The general_personal relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(E_id, charge/hr)

6) opd_doctor relation

E_id→Cons_fees

Key{E_id}

The opd_doctor relation does not have multi-valued attributes hence it is in 1NF.
The opd_doctor relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(E_id, cons_fee)

7) duty_assigned relation

E_id, days→start_shift
E_id, days→end_shift

key{E_id,days}

The duty_assigned relation does not have multi-valued attributes hence it is in 1NF.
The duty_assigned relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So

R(E_id, days, start_shift, end_shift)

8) duty_attendance relation

E_id,dates→swipe_in
E_id,dates→swipe_out

key{E_id,dates}

The duty_attendance relation does not have multi-valued attributes hence it is in 1NF.
The duty_attendance relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(E_id,dates,swipe_in, swipe_out)

9) patient_medical_history relation

p_id→medical_history

key{p_id}

The patient_medical_history relation does not have multi-valued attributes hence it is in 1NF.
The patient_medical_history relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(p_id, medical_history)

10) room_details relation

Room_No→Room_type
Room_No→occupied_status
Room_No→cost

key{room_no}

The room_details relation does not have multi-valued attributes hence it is in 1NF.
The room_details relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.

So
R(room_no, room_type, occupied_status, cost)

11) room_type relation

room_type→room_cost

key{room_type}

The room_type relation does not have multi-valued attributes hence it is in 1NF.
The room_type relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(room_type, room_cost)

12) patient_admit relation

P_id,case_id_admit→Admit_Date
P_id,case_id_admit→Discharge_Date
P_id,case_id_admit→R_No

key{P_id,case_id_admit}

The patient_admit relation does not have multi-valued attributes hence it is in 1NF.
The patient_admit relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(case_id_admit, p_id, admit_date, discharge_date, r_no)

13) policy relation

Policy_No→Policy_Name
Policy_No→claim_amt

key{Policy_No}

The policy relation does not have multi-valued attributes hence it is in 1NF.
The policy relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.

The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(Policy_No, Policy_Name, claim_amt)

14) patient_insurance relation

Policy_no→P_id

Key{policy_no, P_id}

The patient_insurance relation does not have multi-valued attributes hence it is in 1NF.
The patient_insurance relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(Policy_No, p_id)

15) provided_healthcare_service relation

Service_id→dept_no
Service_id→service_desc
Service_id→service_charge
Service_id→service_code

Key{Service_id}

The provided_healthcare_service relation does not have multi-valued attributes hence it is in 1NF.
The provided_healthcare_service relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(service_id, dept_no, service_desc, service_charge, service_code)

16) discharge_summary relation

case_id_admit,P_id→admit_diagnosis

Key{case_id_admit, P_id}

The discharge_summary relation does not have multi-valued attributes hence it is in 1NF.
The discharge_summary relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(case_id_admit, p_id, admit_diagnosis)

17) opd_bill relation

case_id_opd, P_id, e_id→opd_charge
case_id_opd, P_id, e_id→diagnosis
case_id_opd, P_id, e_id→bill_date

Key{case_id_opd, P_id, e_id}

The opd_bill relation does not have multi-valued attributes hence it is in 1NF.
The opd_bill relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
The following dependency is an MVD
case_id_opd, P_id, e_id→diagnosis
So
R1(case_id_opd, P_id, e_id, diagnosis)
R2(case_id_opd, P_id, e_id, opd_charge)

18) interim_bill relation

case_id_admit,P_id,bill_id→room_cost
case_id_admit,P_id,bill_id→service_charge
case_id_admit,P_id,bill_id→opd_charge

key{P_id,bill_id,case_id_admit}

The interim_bill relation does not have multi-valued attributes hence it is in 1NF.
The interim_bill relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(case_id_admit, bill_id, p_id, room_cost, service_charge, opd_charge)

19) total_bill relation

case_id_admit,P_id→policy_no
case_id_admit,P_id→claim_accepted
case_id_admit,P_id→total_charge
case_id_admit,P_id→payable_amt

key{P_id, case_id_admit}

The total_bill relation does not have multi-valued attributes hence it is in 1NF.
The total_bill relation does not have partial dependency hence it is in 2NF.
Additionally, we do not have a transitive dependency, and hence it is in 3NF.
Since we do not have a mixed partial dependency, hence it is also in BCNF.
The FD minimal set satisfies all BCNF requirements. So the relation is in BCNF.
So
R(case_id_admit, p_id, policy_no, claim_accepted, total_charge, payable_amt)

5. Indexing

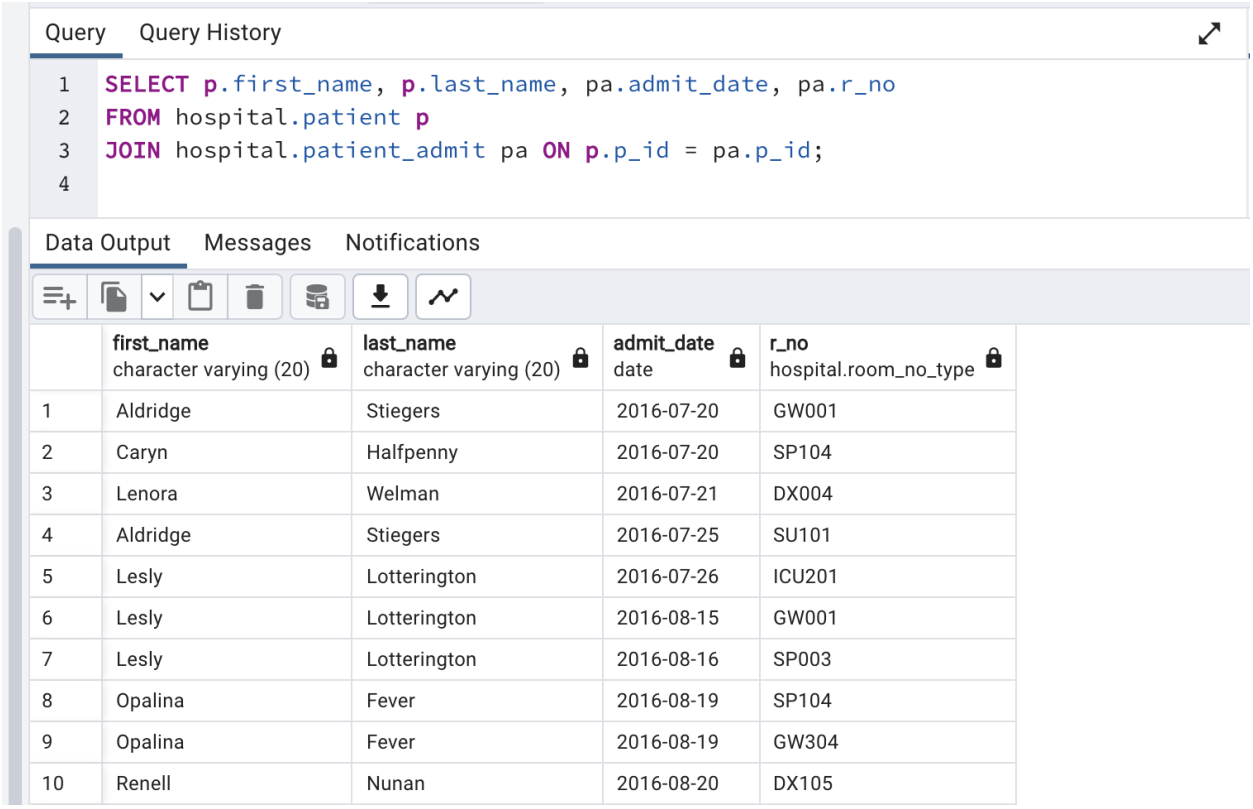
The larger dataset increased the time required to run our queries. The sample dataset, being smaller was performing well, but this delay was not very significant so we did not employ any indexing in our database. If our database had more information, the delay in running queries would be profound in which case employing indexing can enhance query performance, minimize storage demands, and ensure data consistency. But on the other hand, by using indexing, the update and delete queries may take a lot of time to run as the indices will have to be updated every time updation and deletion commands are run.

6. Queries

6.1 Select queries

1. Get the details of all patients admitted to the hospital along with their admission date and room number.

   SELECT p.first_name, p.last_name, pa.admit_date, pa.r_no
   FROM hospital.patient p
   JOIN hospital.patient_admit pa ON p.p_id = pa.p_id;



Query | Query History

```
1  SELECT p.first_name, p.last_name, pa.admit_date, pa.r_no
2  FROM hospital.patient p
3  JOIN hospital.patient_admit pa ON p.p_id = pa.p_id;
4
```

Data Output | Messages | Notifications

	first_name character varying (20)	last_name character varying (20)	admit_date date	r_no hospital.room_no_type
1	Aldridge	Stiegers	2016-07-20	GW001
2	Caryn	Halfpenny	2016-07-20	SP104
3	Lenora	Welman	2016-07-21	DX004
4	Aldridge	Stiegers	2016-07-25	SU101
5	Lesly	Lotterington	2016-07-26	ICU201
6	Lesly	Lotterington	2016-08-15	GW001
7	Lesly	Lotterington	2016-08-16	SP003
8	Opalina	Fever	2016-08-19	SP104
9	Opalina	Fever	2016-08-19	GW304
10	Renell	Nunan	2016-08-20	DX105

2. To retrieve the admit diagnosis and patient details for a given case ID:

   SELECT ds.admit_diagnosis, p.first_name, p.last_name, p.gender, p.phone_number
   FROM hospital.discharge_summary ds
   INNER JOIN hospital.patient p ON ds.p_id = p.p_id
   WHERE ds.case_id_admit = 99;

Query | Query History

```
SELECT ds.admit_diagnosis, p.first_name, p.last_name, p.gender, p.phone_number
FROM hospital.discharge_summary ds
INNER JOIN hospital.patient p ON ds.p_id = p.p_id
WHERE ds.case_id_admit = 99;
```

Data Output | Messages | Notifications

	admit_diagnosis character varying (100)	first_name character varying (20)	last_name character varying (20)	gender character (1)	phone_number character varying (15)
1	Premature Birth	Stace	Buey	F	-3782

3. To retrieve the details of an employee by their ID:

   SELECT e_type, first_name, last_name, dob, address, gender, phone_number,
   emergency_contact, date_of_joining, date_of_resigning
   FROM hospital.employee where e_id = 18 ;

4. To retrieve the policy details for a given patient:

   SELECT pol.policy_no, pol.policy_name, pol.claim_amt
   FROM hospital.patient_insurance pi
   INNER JOIN hospital.policy pol ON pi.policy_no = pol.policy_no
   WHERE pi.p_id = 12;

Query Query History

```
1  SELECT pol.policy_no, pol.policy_name, pol.claim_amt
2  FROM hospital.patient_insurance pi
3  INNER JOIN hospital.policy pol ON pi.policy_no = pol.policy_no
4  WHERE pi.p_id = 12;
5
```

Data Output Messages Notifications

	policy_no [PK] character varying (15)	policy_name character varying (80)	claim_amt money
1	837898294	SBI GENERAL INSURANCE CO LTD	$250,000.00

5. To retrieve the medical history of a given patient:

SELECT medical_history
FROM hospital.patient_medical_history
WHERE p_id = 18;

Query Query History

```
1  SELECT medical_history
2  FROM hospital.patient_medical_history
3  WHERE p_id = 18;
4
```

Data Output Messages Notifications

	medical_history character varying (50)
1	high blood pressure

6.2 Insert queries

1. Query - Insert a new patient:

Before running the sql queries

Query | Query History

```
1  select * from hospital.patient;
2
```

Data Output | Messages | Notifications

	(20)	last_name character varying (20)	gender character (1)	phone_number character varying (15)	dob date	emergency_contact character varying (15)	address character varying (80)	email character varying (50)
79		Kalb	M	-6981	1994-12-24	-3632	5 Express Crossing	[null]
80		Quilty	M	-225	1985-12-10	-1292	448 Weeping Birch Terrace	[null]
81		Stiegers	M	-7178	2003-12-10	-5305	22 Shopko Street	[null]
82		Halfpenny	F	-2553	2003-12-10	-7297	2641 Oak Road	[null]
83		Rodman	M	-5603	1971-10-11	-3549	76 Sycamore Park	[null]
84		Welman	F	-5486	1994-12-24	-7272	38232 Magdeline Park	[null]
85		Rigts	F	-7404	1961-12-28	-8630	3 4th Plaza	[null]
86		Lotterington	F	-5230	1959-12-18	-2212	12493 Beilfuss Street	[null]
87		Krysztofowicz	M	-4034	1973-12-11	-1161	880 Morrow Terrace	[null]
88		Fever	F	-9748	2015-12-16	-1289	8 Sage Lane	[null]
89		Paulino	F	-4166	2016-10-10	-7897	8594 Farwell Hill	[null]
90		Nunan	F	-2454	1975-12-14	-9772	4913 Reindahl Parkway	[null]
91		Rignold	F	-4782	1978-10-12	-7813	0 Corben Street	[null]
92		Healeas	M	-7344	2015-12-10	-3460	656 Bobwhite Plaza	[null]
93		Stoffler	M	-741	1990-10-12	-727	5900 Ramsey Trail	[null]
94		Scotsbrook	F	-2087	2014-12-11	-2472	66534 Maryland Plaza	[null]
95		Scapelhorn	F	-2880	2006-11-12	-7110	98240 Fisk Point	[null]
96		Wilmut	F	-5037	2009-11-19	-7910	5 Arapahoe Circle	[null]
97		Essery	M	-3294	1970-12-13	-7312	00 Shasta Place	[null]
98		Beggini	F	-4946	1967-12-12	-10215	321 Pond Road	[null]
99		Franceschi	M	-4150	1951-12-14	-2118	803 Old Gate Plaza	[null]
100		Avo	F	-7703	1960-12-12	-6024	279 Oxford Lane	[null]

Total rows: 100 of 100 | Query complete 00:00:00.071 | Ln 1, Col 31

INSERT INTO hospital.patient (p_id, first_name, last_name, gender, phone_number, dob, emergency_contact, address, email)

VALUES (101, 'John', 'Doe', 'M', '1234567890', '1990-01-01', '9876543210', '123 Main St', 'johndoe@example.com');

Copy | Copy to Query Editor

```
INSERT INTO hospital.patient (p_id, first_name, last_name, gender, phone_num
VALUES (101, 'John', 'Doe', 'M', '1234567890', '1990-01-01', '9876543210', '
```

Messages

```
Query returned successfully in 289 msec.
```

After running the sql query-

Query | Query History

```
1  select * from hospital.patient;
```

Data Output | Messages | Notifications

	(20)	last_name character varying (20)	gender character (1)	phone_number character varying (15)	dob date	emergency_contact character varying (15)	address character varying (80)	email character varying (50)
80		Quilty	M	-225	1985-12-10	-1292	448 Weeping Birch Terrace	[null]
81		Stiegers	M	-7178	2003-12-10	-5305	22 Shopko Street	[null]
82		Halfpenny	F	-2553	2003-12-10	-7297	2641 Oak Road	[null]
83		Rodman	M	-5603	1971-10-11	-3549	76 Sycamore Park	[null]
84		Welman	F	-5486	1994-12-24	-7272	38232 Magdeline Park	[null]
85		Rigts	F	-7404	1961-12-28	-8630	3 4th Plaza	[null]
86		Lotterington	F	-5230	1959-12-18	-2212	12493 Beilfuss Street	[null]
87		Krysztofowicz	M	-4034	1973-12-11	-1161	880 Morrow Terrace	[null]
88		Fever	F	-9748	2015-12-16	-1289	8 Sage Lane	[null]
89		Paulino	F	-4166	2016-10-10	-7897	8594 Farwell Hill	[null]
90		Nunan	F	-2454	1975-12-14	-9772	4913 Reindahl Parkway	[null]
91		Rignold	F	-4782	1978-10-12	-7813	0 Corben Street	[null]
92		Healeas	M	-7344	2015-12-10	-3460	656 Bobwhite Plaza	[null]
93		Stoffler	M	-741	1990-10-12	-727	5900 Ramsey Trail	[null]
94		Scotsbrook	F	-2087	2014-12-11	-2472	66534 Maryland Plaza	[null]
95		Scapelhorn	F	-2880	2006-11-12	-7110	98240 Fisk Point	[null]
96		Wilmut	F	-5037	2009-11-19	-7910	5 Arapahoe Circle	[null]
97		Essery	M	-3294	1970-12-13	-7312	00 Shasta Place	[null]
98		Beggini	F	-4946	1967-12-12	-10215	321 Pond Road	[null]
99		Franceschi	M	-4150	1951-12-14	-2118	803 Old Gate Plaza	[null]
100		Avo	F	-7703	1960-12-12	-6024	279 Oxford Lane	[null]
101		Doe	M	1234567890	1990-01-01	9876543210	123 Main St	johndoe@example.com

Total rows: 101 of 101 | Query complete 00:00:00.236 | Ln 1, Col 32

2. Insert a new healthcare service into the provided_Healthcare_service table:

   INSERT INTO hospital.provided_healthcare_service (service_id, service_code, dept_no, service_desc, service_charge)
   VALUES ('GC', '001', 1, 'General Checkup', 50.00);

Query | Query History

```
1  select * from hospital.provided_healthcare_service;
```

Data Output | Messages | Notifications

	service_id [PK] character varying	dept_no smallint	service_desc character varying (50)	service_charge money	service_code character (50)
103	DACRYTMY	13	Dacryocystorhinostomy	$200,000.00	[null]
104	EYELIDSURG	13	Eyelid surgery	$50,000.00	[null]
105	STRBSURG	13	Strabismus surgery	$50,000.00	[null]
106	CATASURG	13	Cataract surgery	$60,000.00	[null]
107	GLAUSURG	13	Glaucoma surgery	$60,000.00	[null]
108	RETSURG	13	Retinal surgery	$50,000.00	[null]
109	CORSURG	13	Corneal surgery	$70,000.00	[null]
110	LASEYESURG	13	Laser eye surgery	$30,000.00	[null]
111	REFRCSURG	13	Refractive surgery	$40,000.00	[null]
112	CANPLSTY	13	Canaloplasty	$60,000.00	[null]
113	VITRECTMY	13	vitrectomy	$70,000.00	[null]
114	OCLPLSURG	13	Oculoplastic surgery	$60,000.00	[null]
115	PEDPSY	14	Pediatric psychiatry	$5,000.00	[null]
116	PEDPC	14	Pediatric primary care	$400.00	[null]
117	PEDR	14	Pediatric rheumatology	$10,000.00	[null]
118	PEDGS	14	Pediatric Genetics services	$5,000.00	[null]
119	INF	15	Influenza	$300.00	[null]
120	FEV	15	Fever	$300.00	[null]
121	MIG	15	Migraines	$300.00	[null]
122	COU	15	Cough/Sore throat	$300.00	[null]
123	DRH	15	Diarrhea	$300.00	[null]
124	VOM	15	Vomiting	$300.00	[null]

Total rows: 124 of 124 | Query complete 00:00:00.247 | Ln 1, Col 52

Query run -

Query | Query History

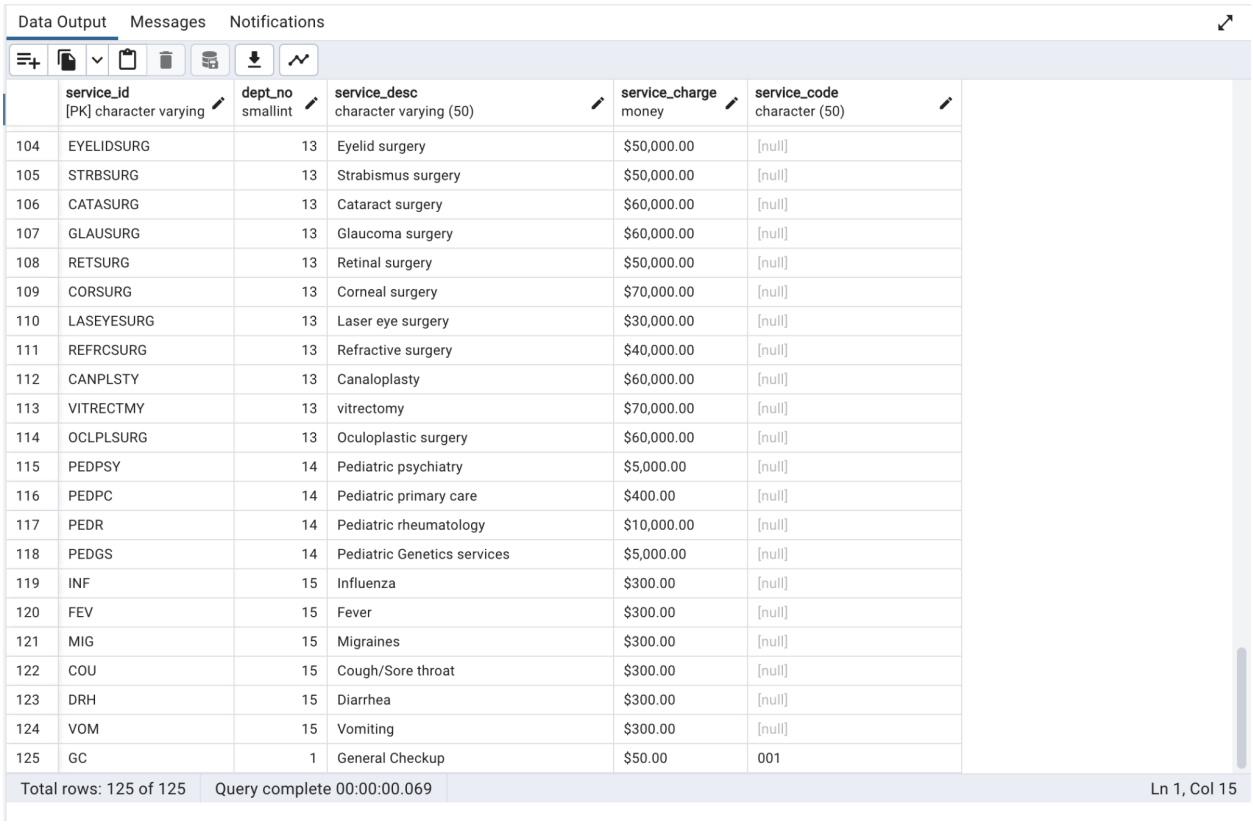
```
1  INSERT INTO hospital.provided_healthcare_service (service_id, service_code, dept_no, service_desc, service_charge)
2  VALUES ('GC', '001', 1, 'General Checkup', 50.00);
3
```

Data Output | Messages | Notifications

```
INSERT 0 1

Query returned successfully in 142 msec.
```

After running the query -



Data Output | Messages | Notifications

	service_id [PK] character varying	dept_no smallint	service_desc character varying (50)	service_charge money	service_code character (50)
104	EYELIDSURG	13	Eyelid surgery	$50,000.00	[null]
105	STRBSURG	13	Strabismus surgery	$50,000.00	[null]
106	CATASURG	13	Cataract surgery	$60,000.00	[null]
107	GLAUSURG	13	Glaucoma surgery	$60,000.00	[null]
108	RETSURG	13	Retinal surgery	$50,000.00	[null]
109	CORSURG	13	Corneal surgery	$70,000.00	[null]
110	LASEYESURG	13	Laser eye surgery	$30,000.00	[null]
111	REFRCSURG	13	Refractive surgery	$40,000.00	[null]
112	CANPLSTY	13	Canaloplasty	$60,000.00	[null]
113	VITRECTMY	13	vitrectomy	$70,000.00	[null]
114	OCLPLSURG	13	Oculoplastic surgery	$60,000.00	[null]
115	PEDPSY	14	Pediatric psychiatry	$5,000.00	[null]
116	PEDPC	14	Pediatric primary care	$400.00	[null]
117	PEDR	14	Pediatric rheumatology	$10,000.00	[null]
118	PEDGS	14	Pediatric Genetics services	$5,000.00	[null]
119	INF	15	Influenza	$300.00	[null]
120	FEV	15	Fever	$300.00	[null]
121	MIG	15	Migraines	$300.00	[null]
122	COU	15	Cough/Sore throat	$300.00	[null]
123	DRH	15	Diarrhea	$300.00	[null]
124	VOM	15	Vomiting	$300.00	[null]
125	GC	1	General Checkup	$50.00	001

Total rows: 125 of 125 | Query complete 00:00:00.069 | Ln 1, Col 15

6.3 Update queries

1. Update the email address of a patient with ID 1:s

Before

Query | Query History

```
1  select * from hospital.patient;
2
```

Data Output | Messages | Notifications

	p_id [PK] numeric (9)	first_name character varying (20)	last_name character varying (20)	gender character (1)	phone_number character varying (15)	dob date	emergency_contact character varying (15)	address character varying (80)	email character varying (50)
1	1	Melody	Mahony	F	-9247	2014-10-13	-6481	29567 Stone Corner Trail	[null]
2	2	Carey	Anscott	M	-2186	1958-11-16	-819	7513 Reindahl Place	[null]
3	3	Horton	Julien	M	-2218	1982-10-20	-6220	26639 Bashford Trail	[null]
4	4	Inge	Keeffe	F	-9625	1954-11-13	-2802	53654 Forster Hill	[null]
5	5	Wes	Martinson	M	-3535	2002-10-16	-9334	814 Reinke Alley	[null]
6	6	Nil	Petren	M	-767	2014-11-12	-4894	123 Barby Lane	[null]
7	7	Shea	Meekin	M	-2253	2009-11-10	-8535	16 Lighthouse Bay Lane	[null]
8	8	Mirna	Rabb	F	-7889	1990-12-19	-9806	2419 1st Hill	[null]
9	9	Dannel	Cicculi	M	-1639	1997-10-17	-9226	9 Garrison Park	[null]
10	10	Pennie	Stillgoe	M	-6860	2001-12-19	-10217	8 Merchant Road	[null]
11	11	Maxwell	Gronaller	M	-1264	1982-11-10	-8223	0815 Spaight Trail	[null]
12	12	Lenka	Cauldfield	F	-8420	1985-11-10	-6874	8 Magdeline Court	[null]
13	13	Lucais	Spriggen	M	-10052	1956-10-13	-3341	7 Ridgeway Park	[null]
14	14	Alfonso	Burston	M	105	1964-12-10	-4435	428 Oak Drive	[null]
15	15	Alleyn	Prosser	M	-7440	1975-12-11	-2567	91336 Anthes Parkway	[null]
16	16	Johan	Padgham	M	-1063	1957-10-12	-6633	423 Karstens Court	[null]
17	17	Caryl	Lukock	M	-2498	1990-12-19	-9216	37626 Jackson Alley	[null]
18	18	Malanie	Mountfort	F	-2354	1985-11-11	-1800	170 Mallard Trail	[null]
19	19	Fiona	Sellar	F	-6272	2016-12-19	-5301	8983 Reindahl Avenue	[null]
20	20	Inga	Speariett	F	-2731	2015-10-22	-4203	5961 Londonderry Place	[null]
21	21	Bartholomew	Storey	M	-5601	1967-11-11	-6085	74 Anhalt Way	[null]
22	22	Demott	Porrett	M	-9306	1982-11-10	-9142	6048 Barby Alley	[null]
23	23	Wernher	Keems	M	-2769	1976-11-12	-5600	711 Manitowish Park	[null]

Total rows: 101 of 101 | Query complete 00:00:00.102 | Ln 1, Col 31

Running the query -

Query | Query History

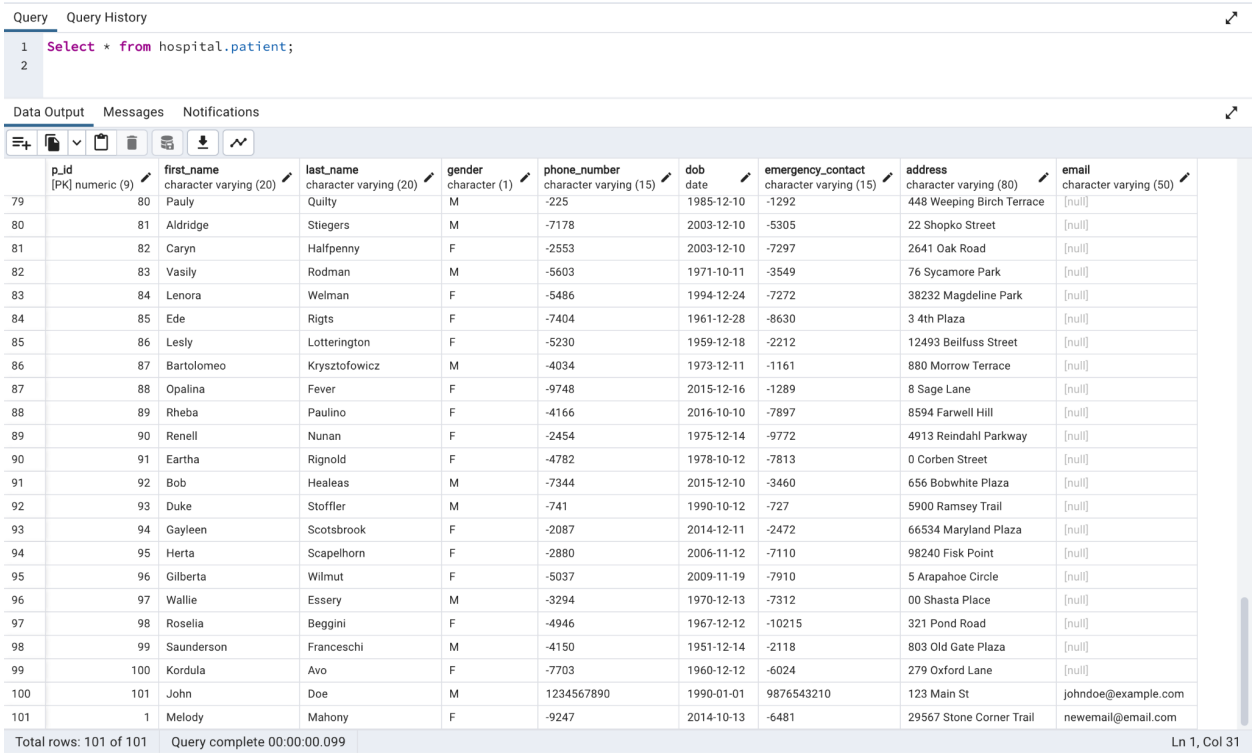
```
1  UPDATE hospital.patient
2  SET email = 'newemail@email.com'
3  WHERE P_id = 1;
```

Data Output | Messages | Notifications

```
UPDATE 1

Query returned successfully in 161 msec.
```

After running the query -



Query | Query History

```
1  Select * from hospital.patient;
2
```

Data Output | Messages | Notifications

	p_id [PK] numeric (9)	first_name character varying (20)	last_name character varying (20)	gender character (1)	phone_number character varying (15)	dob date	emergency_contact character varying (15)	address character varying (80)	email character varying (50)
79	80	Pauly	Quilty	M	-225	1985-12-10	-1292	448 Weeping Birch Terrace	[null]
80	81	Aldridge	Stiegers	M	-7178	2003-12-10	-5305	22 Shopko Street	[null]
81	82	Caryn	Halfpenny	F	-2553	2003-12-10	-7297	2641 Oak Road	[null]
82	83	Vasily	Rodman	M	-5603	1971-10-11	-3549	76 Sycamore Park	[null]
83	84	Lenora	Welman	F	-5486	1994-12-24	-7272	38232 Magdeline Park	[null]
84	85	Ede	Rigts	F	-7404	1961-12-28	-8630	3 4th Plaza	[null]
85	86	Lesly	Lotterington	F	-5230	1959-12-18	-2212	12493 Beilfuss Street	[null]
86	87	Bartolomeo	Krysztofowicz	M	-4034	1973-12-11	-1161	880 Morrow Terrace	[null]
87	88	Opalina	Fever	F	-9748	2015-12-16	-1289	8 Sage Lane	[null]
88	89	Rheba	Paulino	F	-4166	2016-10-10	-7897	8594 Farwell Hill	[null]
89	90	Renell	Nunan	F	-2454	1975-12-14	-9772	4913 Reindahl Parkway	[null]
90	91	Eartha	Rignold	F	-4782	1978-10-12	-7813	0 Corben Street	[null]
91	92	Bob	Healeas	M	-7344	2015-12-10	-3460	656 Bobwhite Plaza	[null]
92	93	Duke	Stoffler	M	-741	1990-10-12	-727	5900 Ramsey Trail	[null]
93	94	Gayleen	Scotsbrook	F	-2087	2014-12-11	-2472	66534 Maryland Plaza	[null]
94	95	Herta	Scapelhorn	F	-2880	2006-11-12	-7110	98240 Fisk Point	[null]
95	96	Gilberta	Wilmut	F	-5037	2009-11-19	-7910	5 Arapahoe Circle	[null]
96	97	Wallie	Essery	M	-3294	1970-12-13	-7312	00 Shasta Place	[null]
97	98	Roselia	Beggini	F	-4946	1967-12-12	-10215	321 Pond Road	[null]
98	99	Saunderson	Franceschi	M	-4150	1951-12-14	-2118	803 Old Gate Plaza	[null]
99	100	Kordula	Avo	F	-7703	1960-12-12	-6024	279 Oxford Lane	[null]
100	101	John	Doe	M	1234567890	1990-01-01	9876543210	123 Main St	johndoe@example.com
101	1	Melody	Mahony	F	-9247	2014-10-13	-6481	29567 Stone Corner Trail	newemail@email.com

Total rows: 101 of 101 | Query complete 00:00:00.099 | Ln 1, Col 31

2. Update the service charge for a healthcare service with ID ‘FA’:

   Before Running the query -

Query | Query History

```
1  Select * from hospital.provided_healthcare_service;
2
```

Data Output | Messages | Notifications

	service_id [PK] character varying	dept_no smallint	service_desc character varying (50)	service_charge money	service_code character (50)
1	FA	1	First aid	$200.00	[null]
2	BLTR	1	Blood transfusion	$20,000.00	[null]
3	HYT	2	Hydrotherapy	$2,000.00	[null]
4	SKG	2	Skin graft	$20,000.00	[null]
5	TET	2	Tetanus shot	$5,000.00	[null]
6	ABL	3	Ablation	$60,000.00	[null]
7	ANGIO	3	Angioplasty	$2,000,000.00	[null]
8	HRTPLNT	3	Heart Transplant	$3,500,000.00	[null]
9	VLVREP	3	Heart Valve Repair	$2,500,000.00	[null]
10	STENT	3	Stenting	$300,000.00	[null]
11	PCMKR	3	Pacemakers	$300,000.00	[null]
12	ANAES	4	Administering anaesthesia	$5,000.00	[null]
13	COMA	4	Medically induced coma	$500,000.00	[null]
14	24x7M	4	24x7 monitoring of vitals	$500.00	[null]
15	DENG	4	Dengue	$1,500.00	[null]
16	TYP	4	Typhoid	$15,000.00	[null]
17	CHO	4	Cholera	$1,500.00	[null]
18	JAUN	4	Jaundice	$1,000.00	[null]
19	REHAB	4	Rehabilitation	$1,000.00	[null]
20	OESENDO	5	Flexible Oesophageal Endoscopy	$100,000.00	[null]
21	PTAUD	5	Pure Tone Audiometry	$80,000.00	[null]
22	FFAUD	5	Free field Audiometry	$80,000.00	[null]
23	CRAUD	5	Conditioned Response Audiometry	$85,000.00	[null]

Total rows: 125 of 125 | Query complete 00:00:00.115 | Ln 1, Col 33

Running the query -

Query | Query History

```
1  UPDATE hospital.provided_healthcare_service
2  SET Service_charge = 150.00
3  WHERE Service_id = 'FA';
```

Data Output | Messages | Notifications

```
UPDATE 1

Query returned successfully in 100 msec.
```

After running the Query -

Query | Query History

```
1  select * from hospital.provided_healthcare_service;
2
```

Data Output | Messages | Notifications

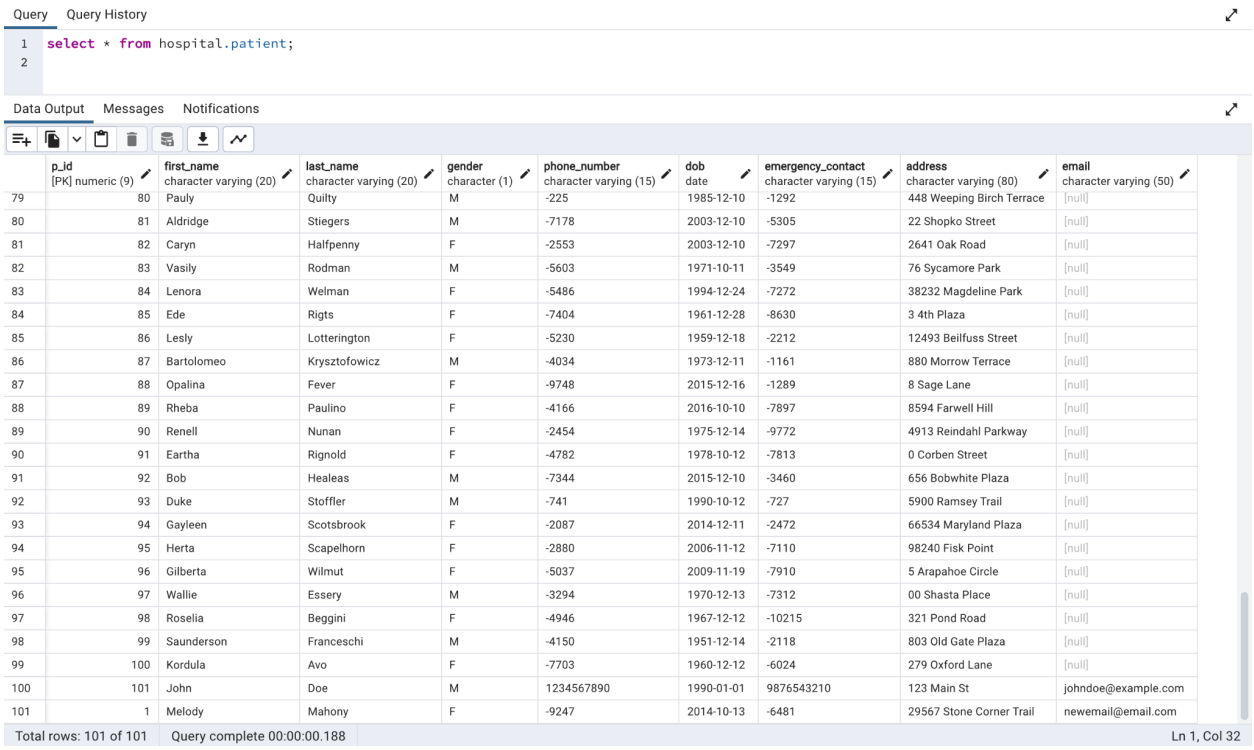
	service_id [PK] character varying	dept_no smallint	service_desc character varying (50)	service_charge money	service_code character (50)
103	EYELIDSURG	13	Eyelid surgery	$50,000.00	[null]
104	STRBSURG	13	Strabismus surgery	$50,000.00	[null]
105	CATASURG	13	Cataract surgery	$60,000.00	[null]
106	GLAUSURG	13	Glaucoma surgery	$60,000.00	[null]
107	RETSURG	13	Retinal surgery	$50,000.00	[null]
108	CORSURG	13	Corneal surgery	$70,000.00	[null]
109	LASEYESURG	13	Laser eye surgery	$30,000.00	[null]
110	REFRCSURG	13	Refractive surgery	$40,000.00	[null]
111	CANPLSTY	13	Canaloplasty	$60,000.00	[null]
112	VITRECTMY	13	vitrectomy	$70,000.00	[null]
113	OCLPLSURG	13	Oculoplastic surgery	$60,000.00	[null]
114	PEDPSY	14	Pediatric psychiatry	$5,000.00	[null]
115	PEDPC	14	Pediatric primary care	$400.00	[null]
116	PEDR	14	Pediatric rheumatology	$10,000.00	[null]
117	PEDGS	14	Pediatric Genetics services	$5,000.00	[null]
118	INF	15	Influenza	$300.00	[null]
119	FEV	15	Fever	$300.00	[null]
120	MIG	15	Migraines	$300.00	[null]
121	COU	15	Cough/Sore throat	$300.00	[null]
122	DRH	15	Diarrhea	$300.00	[null]
123	VOM	15	Vomiting	$300.00	[null]
124	GC	1	General Checkup	$50.00	001
125	FA	1	First aid	$150.00	[null]

Total rows: 125 of 125 | Query complete 00:00:00.079 | Ln 1, Col 15

6.4 Delete queries

1. Delete a patient with ID 101:

   Before Running the query:



Query | Query History

```
1  select * from hospital.patient;
2
```

Data Output | Messages | Notifications

	p_id [PK] numeric (9)	first_name character varying (20)	last_name character varying (20)	gender character (1)	phone_number character varying (15)	dob date	emergency_contact character varying (15)	address character varying (80)	email character varying (50)
79	80	Pauly	Quilty	M	-225	1985-12-10	-1292	448 Weeping Birch Terrace	[null]
80	81	Aldridge	Stiegers	M	-7178	2003-12-10	-5305	22 Shopko Street	[null]
81	82	Caryn	Halfpenny	F	-2553	2003-12-10	-7297	2641 Oak Road	[null]
82	83	Vasily	Rodman	M	-5603	1971-10-11	-3549	76 Sycamore Park	[null]
83	84	Lenora	Welman	F	-5486	1994-12-24	-7272	38232 Magdeline Park	[null]
84	85	Ede	Rigts	F	-7404	1961-12-28	-8630	3 4th Plaza	[null]
85	86	Lesly	Lotterington	F	-5230	1959-12-18	-2212	12493 Beilfuss Street	[null]
86	87	Bartolomeo	Krysztofowicz	M	-4034	1973-12-11	-1161	880 Morrow Terrace	[null]
87	88	Opalina	Fever	F	-9748	2015-12-16	-1289	8 Sage Lane	[null]
88	89	Rheba	Paulino	F	-4166	2016-10-10	-7897	8594 Farwell Hill	[null]
89	90	Renell	Nunan	F	-2454	1975-12-14	-9772	4913 Reindahl Parkway	[null]
90	91	Eartha	Rignold	F	-4782	1978-10-12	-7813	0 Corben Street	[null]
91	92	Bob	Healeas	M	-7344	2015-12-10	-3460	656 Bobwhite Plaza	[null]
92	93	Duke	Stoffler	M	-741	1990-10-12	-727	5900 Ramsey Trail	[null]
93	94	Gayleen	Scotsbrook	F	-2087	2014-12-11	-2472	66534 Maryland Plaza	[null]
94	95	Herta	Scapelhorn	F	-2880	2006-11-12	-7110	98240 Fisk Point	[null]
95	96	Gilberta	Wilmut	F	-5037	2009-11-19	-7910	5 Arapahoe Circle	[null]
96	97	Wallie	Essery	M	-3294	1970-12-13	-7312	00 Shasta Place	[null]
97	98	Roselia	Beggini	F	-4946	1967-12-12	-10215	321 Pond Road	[null]
98	99	Saunderson	Franceschi	M	-4150	1951-12-14	-2118	803 Old Gate Plaza	[null]
99	100	Kordula	Avo	F	-7703	1960-12-12	-6024	279 Oxford Lane	[null]
100	101	John	Doe	M	1234567890	1990-01-01	9876543210	123 Main St	johndoe@example.com
101	1	Melody	Mahony	F	-9247	2014-10-13	-6481	29567 Stone Corner Trail	newemail@email.com

Total rows: 101 of 101 | Query complete 00:00:00.188 | Ln 1, Col 32

   Running the query -

```
DELETE FROM hospital.patient
WHERE p_id = 101;
```

```
Messages
Query returned successfully in 148 msec.
```

   After the query -

Query | Query History

```
1  select * from hospital.patient;
2
```

Data Output | Messages | Notifications

	p_id [PK] numeric (9)	first_name character varying (20)	last_name character varying (20)	gender character (1)	phone_number character varying (15)	dob date	emergency_contact character varying (15)	address character varying (80)	email character varying (50)
79	80	Pauly	Quilty	M	-225	1985-12-10	-1292	448 Weeping Birch Terrace	[null]
80	81	Aldridge	Stiegers	M	-7178	2003-12-10	-5305	22 Shopko Street	[null]
81	82	Caryn	Halfpenny	F	-2553	2003-12-10	-7297	2641 Oak Road	[null]
82	83	Vasily	Rodman	M	-5603	1971-10-11	-3549	76 Sycamore Park	[null]
83	84	Lenora	Welman	F	-5486	1994-12-24	-7272	38232 Magdeline Park	[null]
84	85	Ede	Rigts	F	-7404	1961-12-28	-8630	3 4th Plaza	[null]
85	86	Lesly	Lotterington	F	-5230	1959-12-18	-2212	12493 Beilfuss Street	[null]
86	87	Bartolomeo	Krysztofowicz	M	-4034	1973-12-11	-1161	880 Morrow Terrace	[null]
87	88	Opalina	Fever	F	-9748	2015-12-16	-1289	8 Sage Lane	[null]
88	89	Rheba	Paulino	F	-4166	2016-10-10	-7897	8594 Farwell Hill	[null]
89	90	Renell	Nunan	F	-2454	1975-12-14	-9772	4913 Reindahl Parkway	[null]
90	91	Eartha	Rignold	F	-4782	1978-10-12	-7813	0 Corben Street	[null]
91	92	Bob	Healeas	M	-7344	2015-12-10	-3460	656 Bobwhite Plaza	[null]
92	93	Duke	Stoffler	M	-741	1990-10-12	-727	5900 Ramsey Trail	[null]
93	94	Gayleen	Scotsbrook	F	-2087	2014-12-11	-2472	66534 Maryland Plaza	[null]
94	95	Herta	Scapelhorn	F	-2880	2006-11-12	-7110	98240 Fisk Point	[null]
95	96	Gilberta	Wilmut	F	-5037	2009-11-19	-7910	5 Arapahoe Circle	[null]
96	97	Wallie	Essery	M	-3294	1970-12-13	-7312	00 Shasta Place	[null]
97	98	Roselia	Beggini	F	-4946	1967-12-12	-10215	321 Pond Road	[null]
98	99	Saunderson	Franceschi	M	-4150	1951-12-14	-2118	803 Old Gate Plaza	[null]
99	100	Kordula	Avo	F	-7703	1960-12-12	-6024	279 Oxford Lane	[null]
100	1	Melody	Mahony	F	-9247	2014-10-13	-6481	29567 Stone Corner Trail	newemail@email.com

Total rows: 100 of 100 | Query complete 00:00:00.155 | Ln 2, Col 1

2. Delete a healthcare service with ID 'S003':
Before running the query -

Query | Query History

```
1  select * from hospital.provided_healthcare_service;
2
```

Data Output | Messages | Notifications

	service_id [PK] character varying	dept_no smallint	service_desc character varying (50)	service_charge money	service_code character (50)
106	GLAUSURG	13	Glaucoma surgery	$60,000.00	[null]
107	RETSURG	13	Retinal surgery	$50,000.00	[null]
108	CORSURG	13	Corneal surgery	$70,000.00	[null]
109	LASEYESURG	13	Laser eye surgery	$30,000.00	[null]
110	REFRCSURG	13	Refractive surgery	$40,000.00	[null]
111	CANPLSTY	13	Canaloplasty	$60,000.00	[null]
112	VITRECTMY	13	vitrectomy	$70,000.00	[null]
113	OCLPLSURG	13	Oculoplastic surgery	$60,000.00	[null]
114	PEDPSY	14	Pediatric psychiatry	$5,000.00	[null]
115	PEDPC	14	Pediatric primary care	$400.00	[null]
116	PEDR	14	Pediatric rheumatology	$10,000.00	[null]
117	PEDGS	14	Pediatric Genetics services	$5,000.00	[null]
118	INF	15	Influenza	$300.00	[null]
119	FEV	15	Fever	$300.00	[null]
120	MIG	15	Migraines	$300.00	[null]
121	COU	15	Cough/Sore throat	$300.00	[null]
122	DRH	15	Diarrhea	$300.00	[null]
123	VOM	15	Vomiting	$300.00	[null]
124	GC	1	General Checkup	$50.00	001
125	FA	1	First aid	$150.00	[null]

Total rows: 125 of 125 | Query complete 00:00:00.127 | Ln 1, Col 51

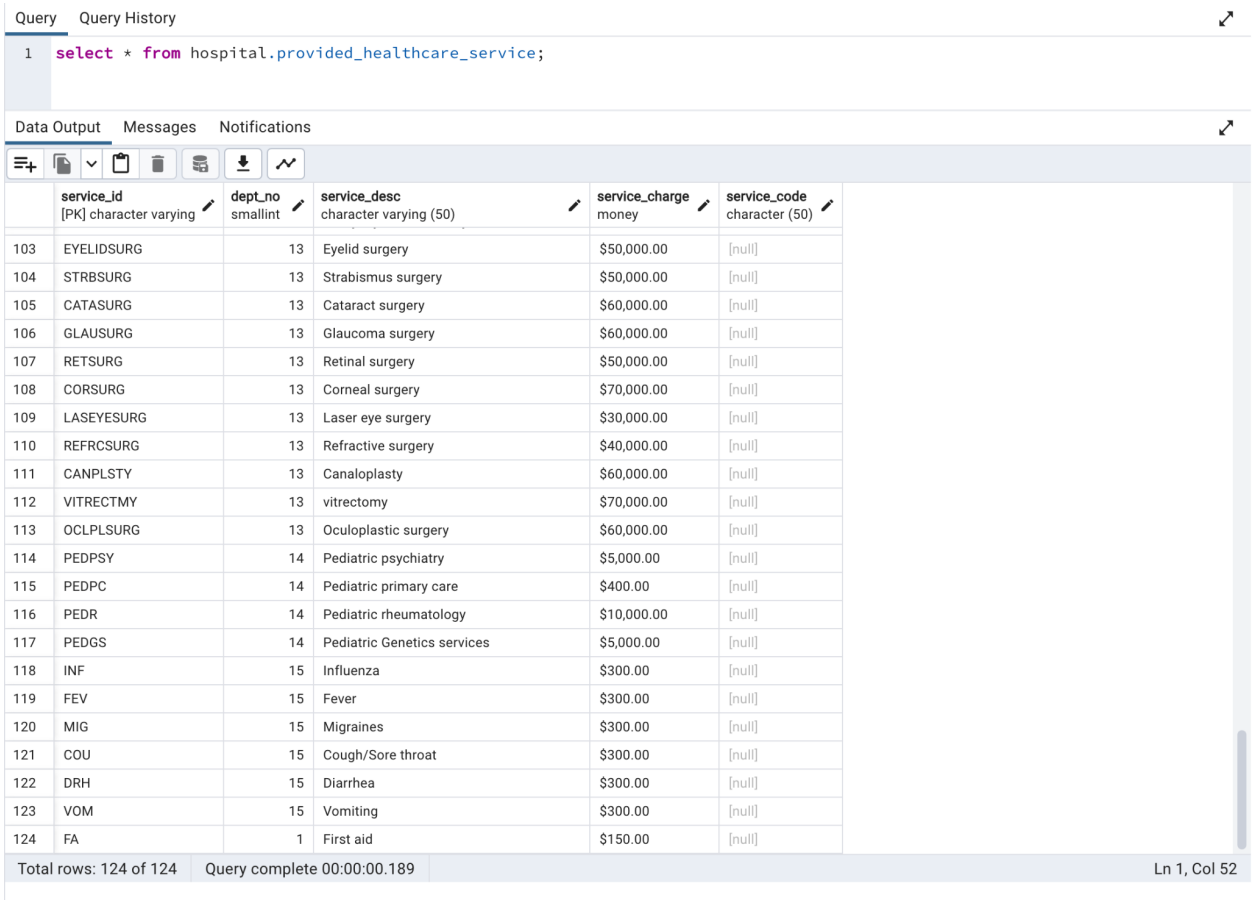
Running the query -

```
DELETE FROM hospital.provided_healthcare_service
WHERE service_id = 'GC';
```

Messages

```
Query returned successfully in 79 msec.
```

After running the query -



```
Query  Query History
1  select * from hospital.provided_healthcare_service;
```

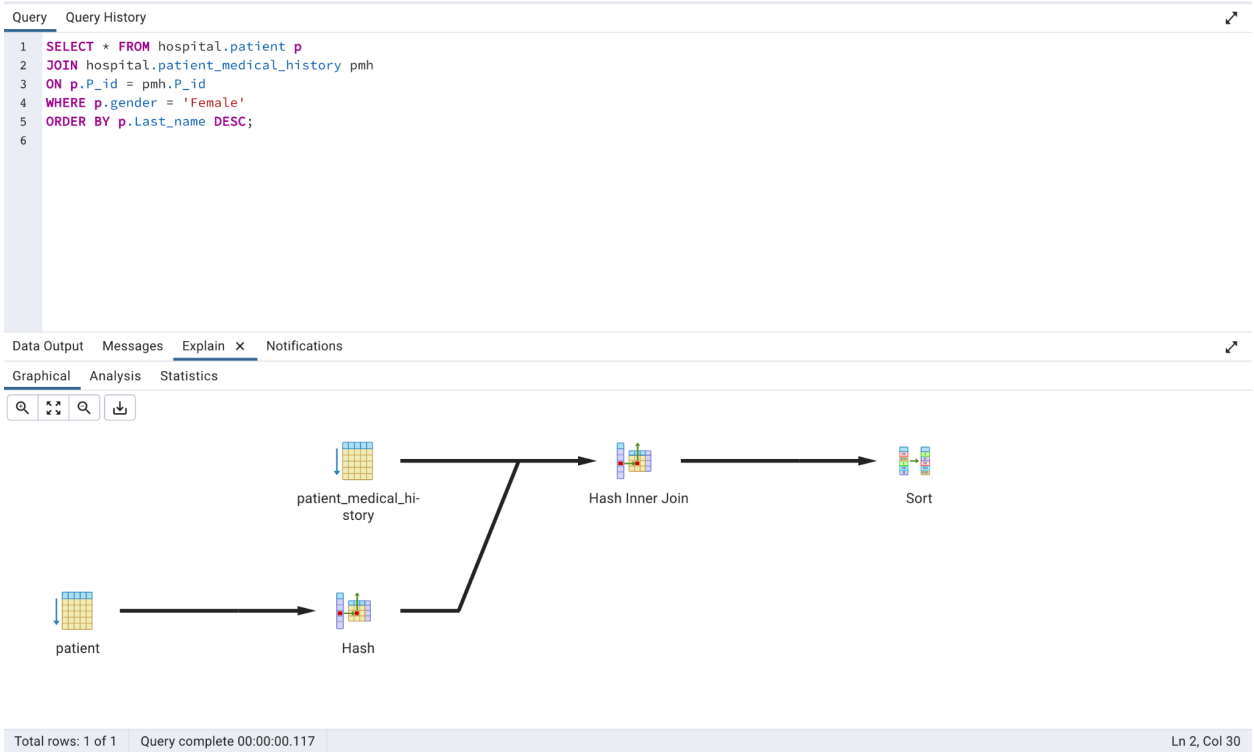
Data Output | Messages | Notifications

	service_id [PK] character varying	dept_no smallint	service_desc character varying (50)	service_charge money	service_code character (50)
103	EYELIDSURG	13	Eyelid surgery	$50,000.00	[null]
104	STRBSURG	13	Strabismus surgery	$50,000.00	[null]
105	CATASURG	13	Cataract surgery	$60,000.00	[null]
106	GLAUSURG	13	Glaucoma surgery	$60,000.00	[null]
107	RETSURG	13	Retinal surgery	$50,000.00	[null]
108	CORSURG	13	Corneal surgery	$70,000.00	[null]
109	LASEYESURG	13	Laser eye surgery	$30,000.00	[null]
110	REFRCSURG	13	Refractive surgery	$40,000.00	[null]
111	CANPLSTY	13	Canaloplasty	$60,000.00	[null]
112	VITRECTMY	13	vitrectomy	$70,000.00	[null]
113	OCLPLSURG	13	Oculoplastic surgery	$60,000.00	[null]
114	PEDPSY	14	Pediatric psychiatry	$5,000.00	[null]
115	PEDPC	14	Pediatric primary care	$400.00	[null]
116	PEDR	14	Pediatric rheumatology	$10,000.00	[null]
117	PEDGS	14	Pediatric Genetics services	$5,000.00	[null]
118	INF	15	Influenza	$300.00	[null]
119	FEV	15	Fever	$300.00	[null]
120	MIG	15	Migraines	$300.00	[null]
121	COU	15	Cough/Sore throat	$300.00	[null]
122	DRH	15	Diarrhea	$300.00	[null]
123	VOM	15	Vomiting	$300.00	[null]
124	FA	1	First aid	$150.00	[null]

Total rows: 124 of 124 | Query complete 00:00:00.189 | Ln 1, Col 52

7. Problematic Queries with Explanation tool

1.



Execution Cost: The cost of this query can be high, especially if there are many rows in the Patient and Patient_med_history tables. The join operation can also be expensive, particularly if there are no indexes on the P_id column.

Improvement Plan:
1. Add an index on the P_id column of both the Patient and Patient_med_history tables.

2. Restrict the number of columns being selected by specifying only the necessary columns instead of using SELECT *.
3. Use a WHERE clause to filter the results before joining the tables.
4. Consider using a subquery to retrieve the necessary data instead of joining the tables.

2.

Query | Query History

```
1  SELECT p.p_id, p.first_name, p.last_name, pmh.medical_history
2  FROM hospital.patient p
3  LEFT JOIN hospital.patient_medical_history pmh
4  ON p.P_id = pmh.P_id
5  WHERE pmh.medical_history IS NULL;
6
```

Data Output | Messages | Explain | Notifications

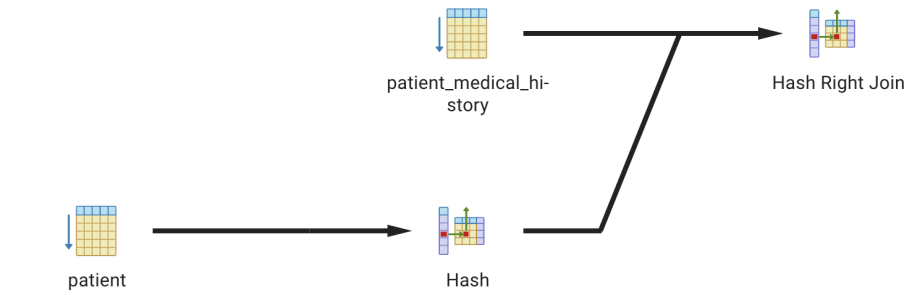
	p_id numeric (9)	first_name character varying (20)	last_name character varying (20)	medical_history character varying (50)
1	12	Lenka	Cauldfield	[null]
2	23	Wernher	Keems	[null]
3	56	Ingrid	Ousby	[null]
4	41	Josselyn	Inskipp	[null]
5	32	Cordell	Mixter	[null]
6	93	Duke	Stoffler	[null]
7	20	Inga	Speariett	[null]
8	7	Shea	Meekin	[null]
9	97	Wallie	Essery	[null]
10	52	Prudi	Colbron	[null]
11	44	Gherardo	Hoodspeth	[null]
12	49	Inglebert	Brooks	[null]
13	88	Opalina	Fever	[null]
14	1	Melody	Mahony	[null]
15	4	Inge	Keeffe	[null]
16	36	Delia	Danilovic	[null]
17	16	Johan	Padgham	[null]

Query Query History

```
1  SELECT p.p_id, p.first_name, p.last_name, pmh.medical_history
2  FROM hospital.patient p
3  LEFT JOIN hospital.patient_medical_history pmh
4  ON p.P_id = pmh.P_id
5  WHERE pmh.medical_history IS NULL;
6
7
```

Data Output Messages Explain ✕ Notifications

Graphical Analysis Statistics



Execution Cost: This query can be expensive, particularly if there are many rows in the Patient and Patient_med_history tables, and there are no indexes on the P_id column.

Improvement Plan:
1. Add an index on the P_id column of both the Patient and Patient_med_history tables.
2. Use a subquery to retrieve the necessary data instead of joining the tables.
3. Restrict the number of columns being selected by specifying only the necessary columns instead of using SELECT *.
4. Consider using a temporary table to store the intermediate results.

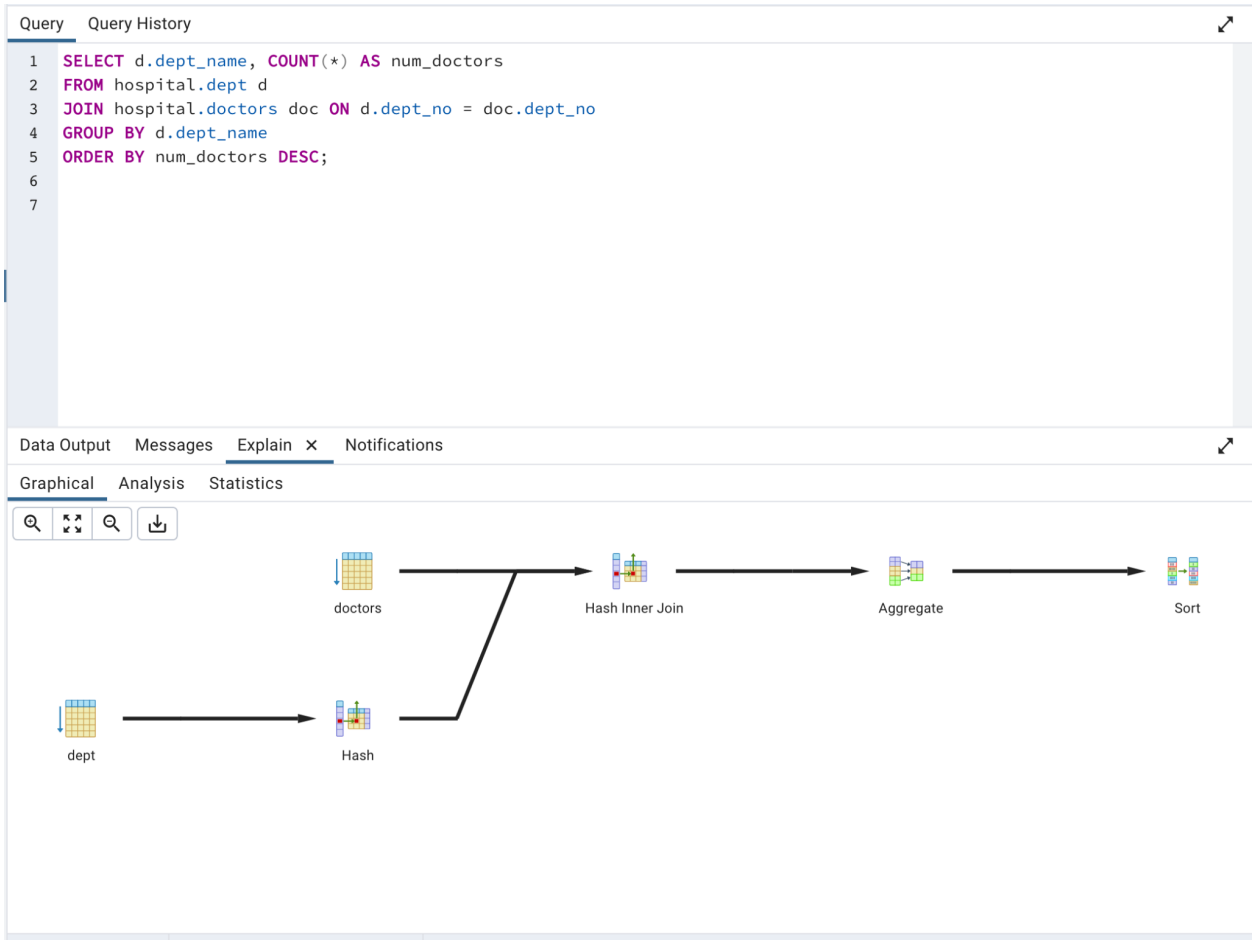
3.

Query Query History

```
1  SELECT d.dept_name, COUNT(*) AS num_doctors
2  FROM hospital.dept d
3  JOIN hospital.doctors doc ON d.dept_no = doc.dept_no
4  GROUP BY d.dept_name
5  ORDER BY num_doctors DESC;
6
```

Data Output Messages Explain ✕ Notifications

dept_name character varying (25)	num_doctors bigint



Execution Cost: This query can be expensive, particularly if the Department and Doctor tables are large, and there are no indexes on the Dept_no column.

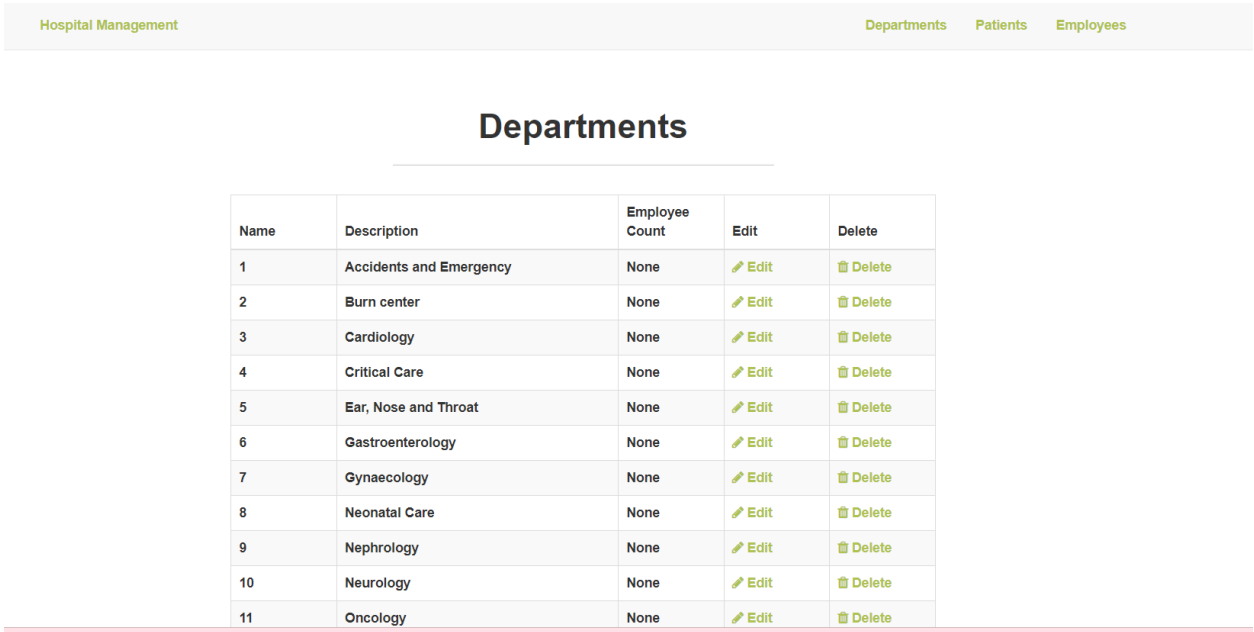
Improvement Plan:
1. Add an index on the Dept_no column of the Department and Doctor tables.
2. Restrict the number of columns being selected by specifying only the necessary columns instead of using SELECT *.
3. Consider using a materialized view to pre-aggregate the data.
4. Use a CTE (Common Table Expression) to calculate the aggregate values before joining the tables.

8. Bonus Question

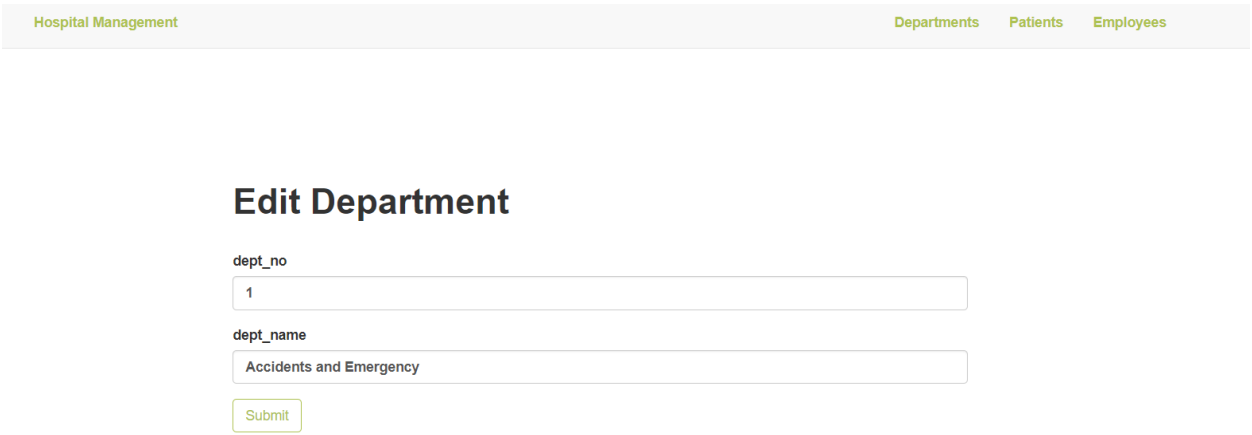
WEBSITE

8.1 Department relation

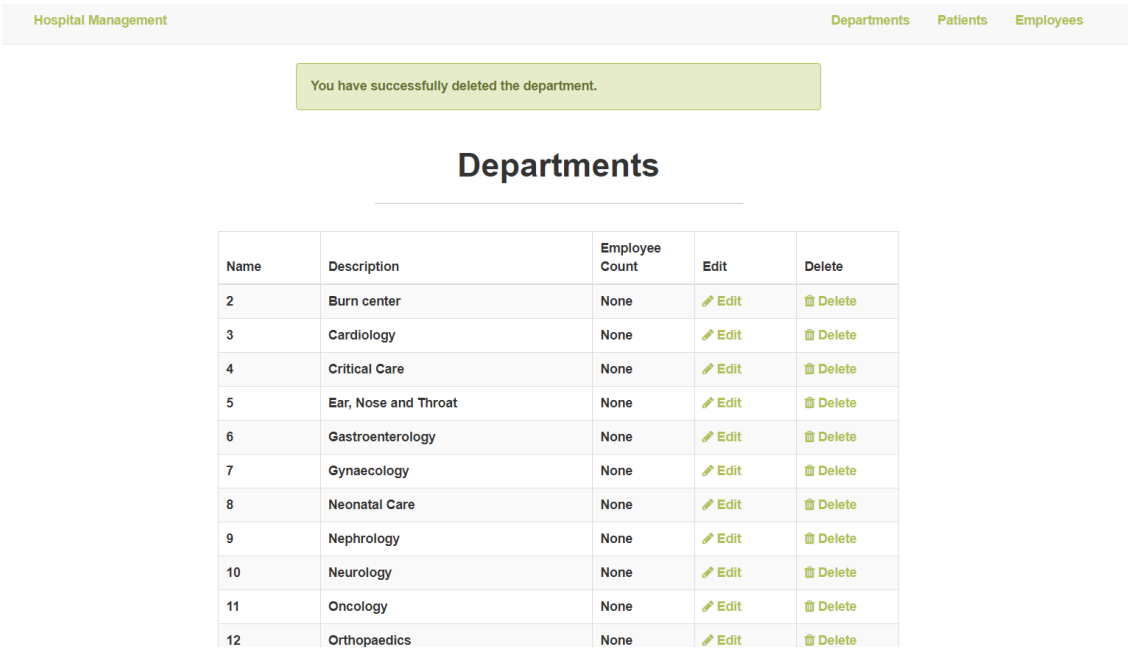
This tab presents the department relation present in our database.



We can perform ‘Edit’ command on any tuple by clicking the ‘Edit’ button next to the said tuple. This allows us to edit the various attributes like dept_no and dept_name in existing tuples. All these changes are also reflected in the database in PostgreSQL.

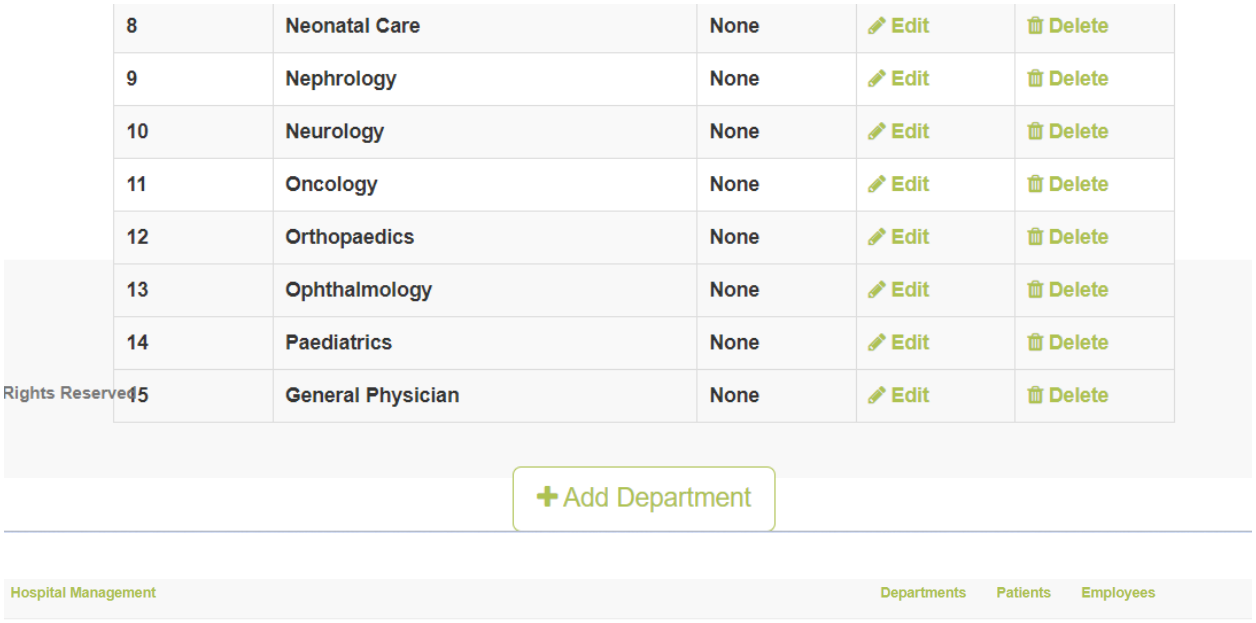


We can perform ‘Delete’ command on any tuple by clicking the ‘Delete’ button next to the said tuple. This allows us to delete the tuple. All these changes are also reflected in the database in PostgreSQL.



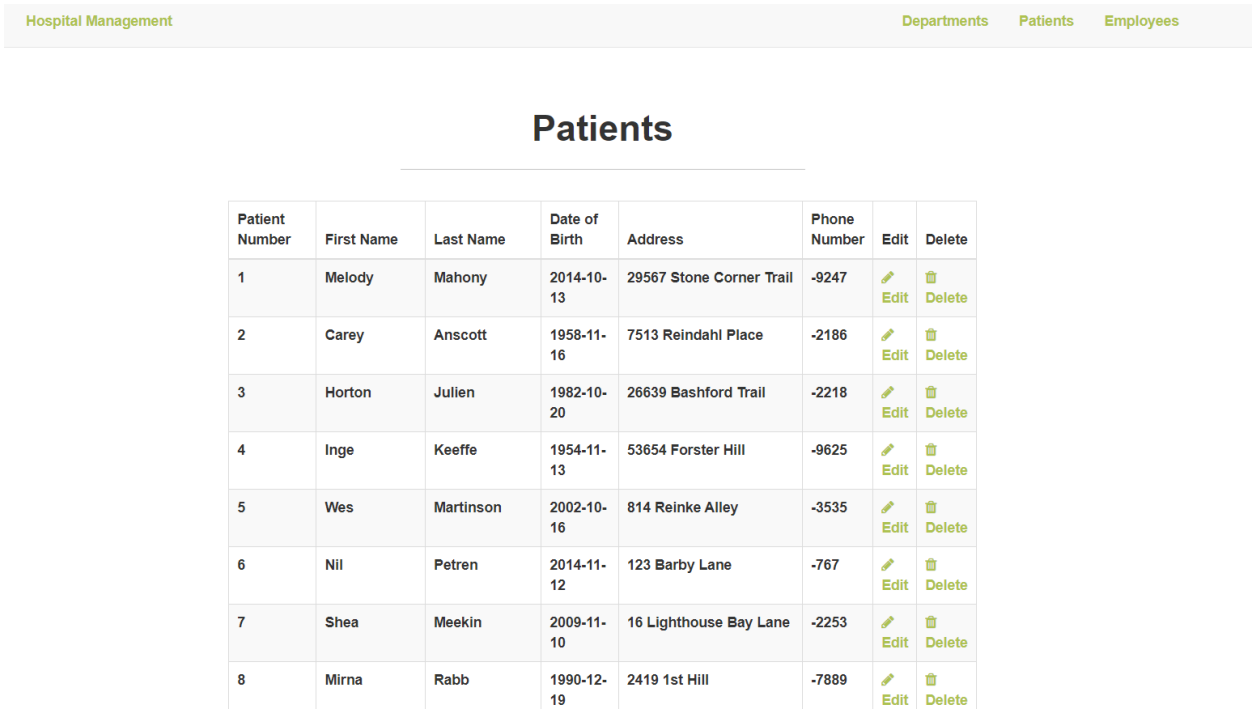
Name	Description	Employee Count	Edit	Delete
2	Burn center	None	Edit	Delete
3	Cardiology	None	Edit	Delete
4	Critical Care	None	Edit	Delete
5	Ear, Nose and Throat	None	Edit	Delete
6	Gastroenterology	None	Edit	Delete
7	Gynaecology	None	Edit	Delete
8	Neonatal Care	None	Edit	Delete
9	Nephrology	None	Edit	Delete
10	Neurology	None	Edit	Delete
11	Oncology	None	Edit	Delete
12	Orthopaedics	None	Edit	Delete

We can perform ‘Insert’ command on any tuple by clicking ‘Add department’ at the end of the relation table. This allows us to add a new tuple to the department relation. All these changes are also reflected in the database in PostgreSQL.

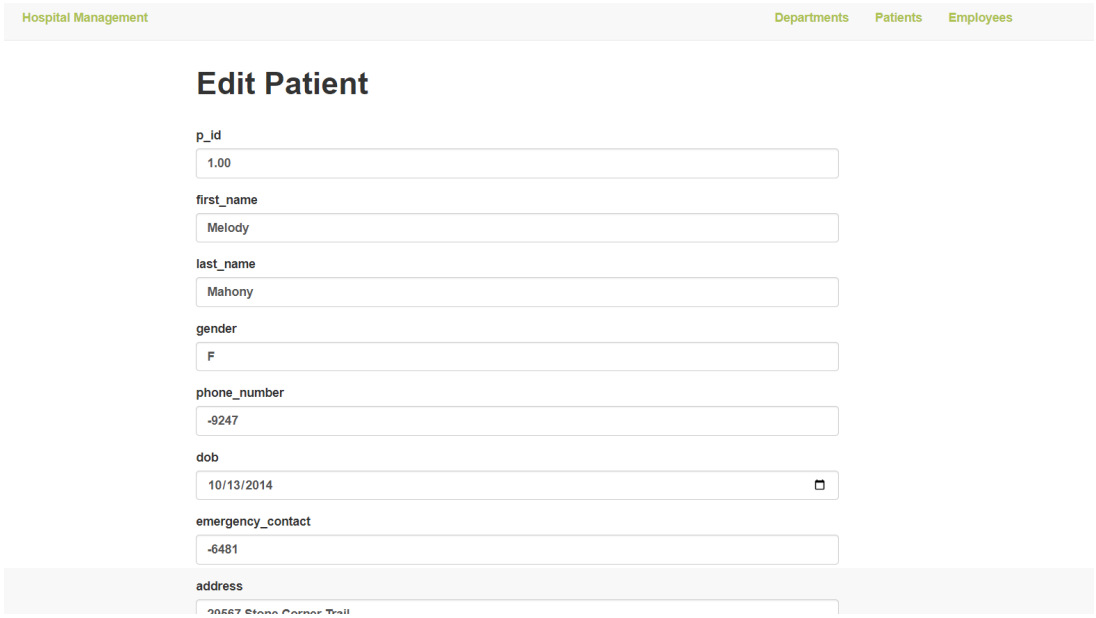


8.2 Patient relation

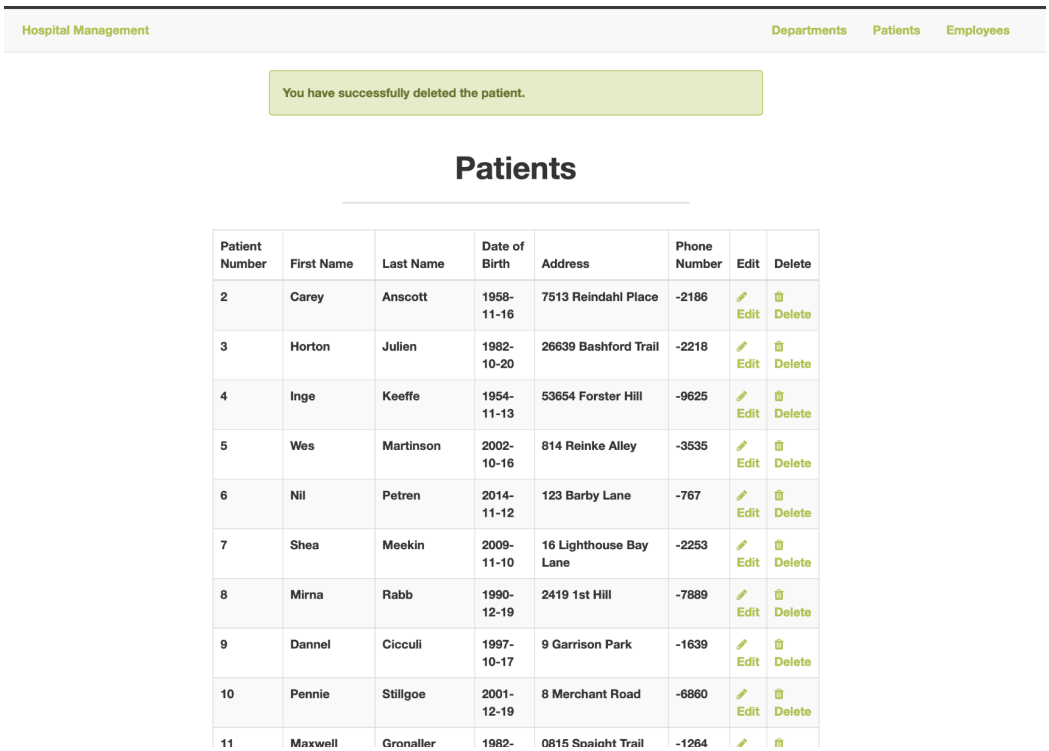
This tab presents the patient relation present in our database.



We can perform 'Edit' command on any tuple by clicking the 'Edit' button next to the said tuple. This allows us to edit the various attributes like p_id, name, gender, phone-number, dob, emergency contact and address in existing tuples. All these changes are also reflected in the database in PostgreSQL.



We can perform ‘Delete’ command on any tuple by clicking the ‘Delete’ button next to the said tuple. This allows us to delete the tuple. All these changes are also reflected in the database in PostgreSQL.



Patient Number	First Name	Last Name	Date of Birth	Address	Phone Number	Edit	Delete
2	Carey	Anscott	1958-11-16	7513 Reindahl Place	-2186	Edit	Delete
3	Horton	Julien	1982-10-20	26639 Bashford Trail	-2218	Edit	Delete
4	Inge	Keeffe	1954-11-13	53654 Forster Hill	-9625	Edit	Delete
5	Wes	Martinson	2002-10-16	814 Reinke Alley	-3535	Edit	Delete
6	Nil	Petren	2014-11-12	123 Barby Lane	-767	Edit	Delete
7	Shea	Meekin	2009-11-10	16 Lighthouse Bay Lane	-2253	Edit	Delete
8	Mirna	Rabb	1990-12-19	2419 1st Hill	-7889	Edit	Delete
9	Dannel	Cicculi	1997-10-17	9 Garrison Park	-1639	Edit	Delete
10	Pennie	Stillgoe	2001-12-19	8 Merchant Road	-6860	Edit	Delete
11	Maxwell	Gronaller	1982-	0815 Spaight Trail	-1264		

We can perform ‘Insert’ command on any tuple by clicking ‘Add Patient’ at the end of the relation table. This allows us to add a new tuple to the patient relation. All these changes are also reflected in the database in PostgreSQL.

Hospital Management

Departments Patients Employees

91	Eartha	Rignold	1978-10-12	0 Corben Street	-4782	Edit	Delete
92	Bob	Healeas	2015-12-10	656 Bobwhite Plaza	-7344	Edit	Delete
93	Duke	Stoffler	1990-10-12	5900 Ramsey Trail	-741	Edit	Delete
94	Gayleen	Scotsbrook	2014-12-11	66534 Maryland Plaza	-2087	Edit	Delete
95	Herta	Scapelhorn	2006-11-12	98240 Fisk Point	-2880	Edit	Delete
96	Gilberta	Wilmut	2009-11-19	5 Arapahoe Circle	-5037	Edit	Delete
97	Wallie	Essery	1970-12-13	00 Shasta Place	-3294	Edit	Delete
98	Roselia	Beggini	1967-12-12	321 Pond Road	-4946	Edit	Delete
99	Saunderson	Franceschi	1951-12-14	803 Old Gate Plaza	-4150	Edit	Delete
100	Kordula	Avo	1960-12-12	279 Oxford Lane	-7703	Edit	Delete

+ Add Patient

Hospital Management

Departments Patients Employees

Add Patient

p_id

first_name

last_name

gender

phone_number

dob

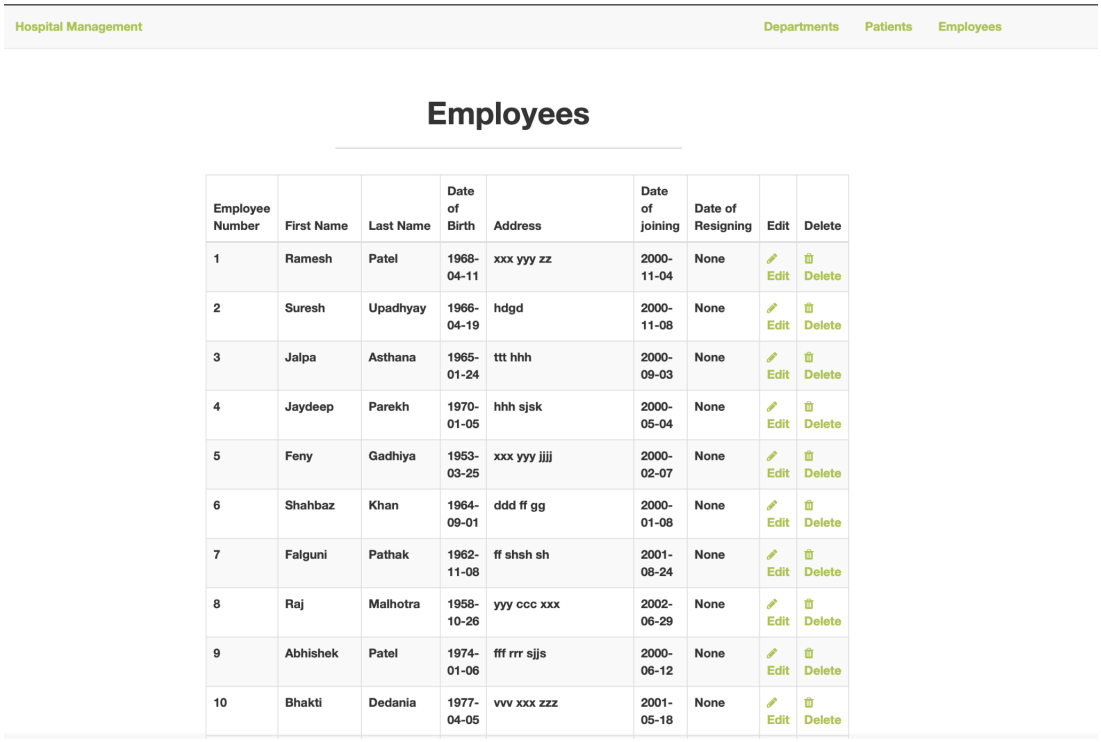
mm/dd/yyyy

emergency_contact

address

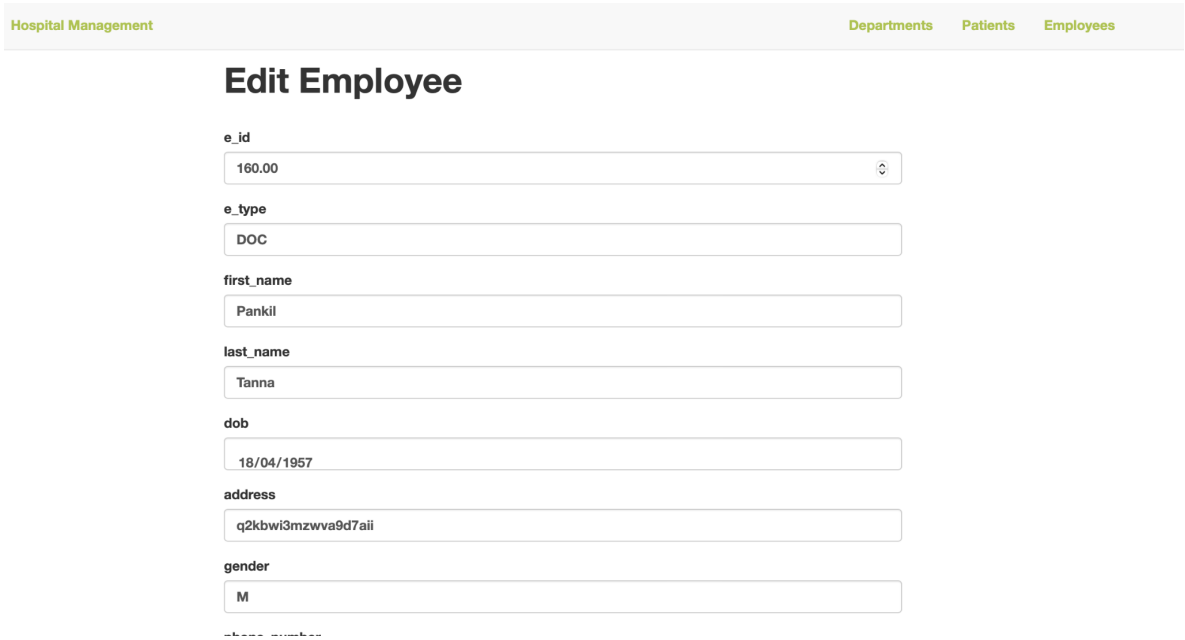
8.3 Employee relation

This tab presents the employee relation present in our database.

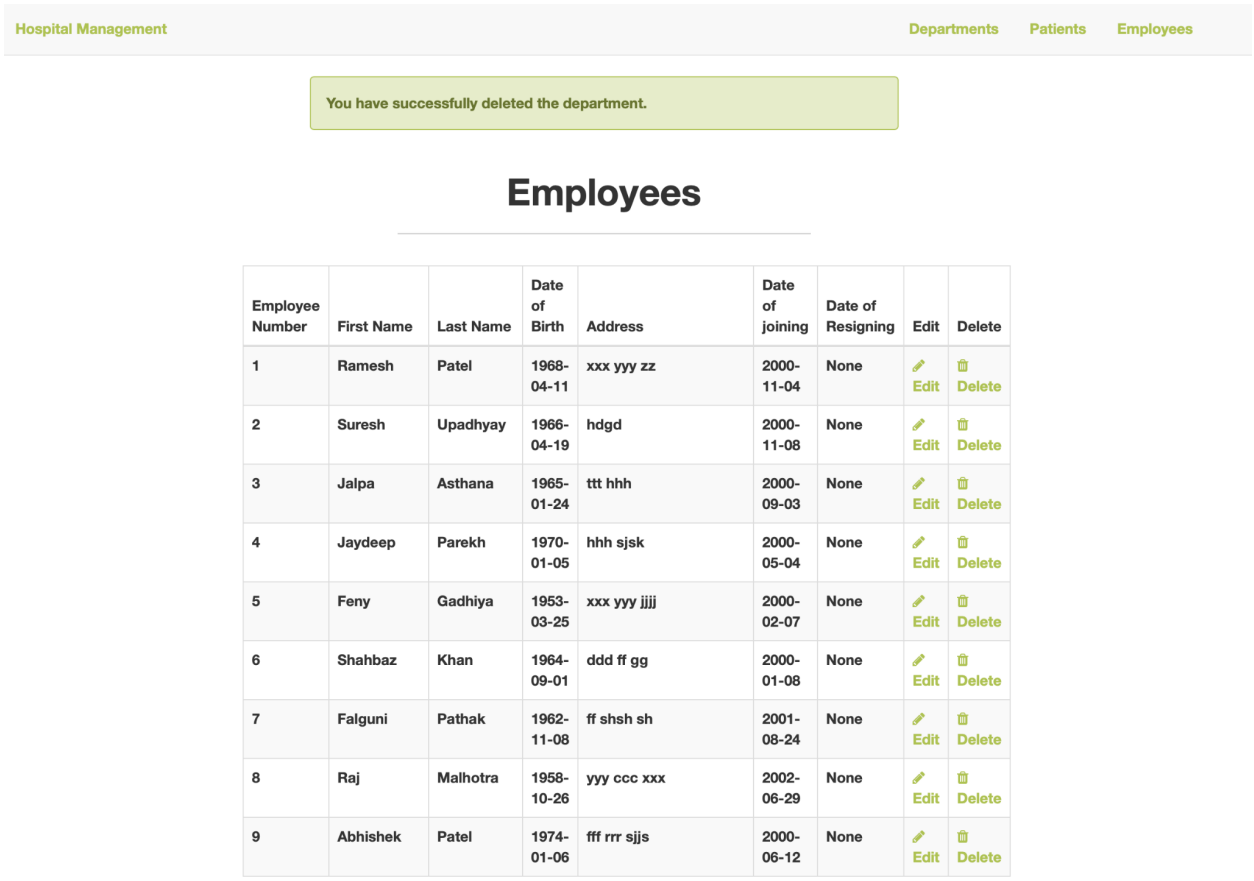


Employee Number	First Name	Last Name	Date of Birth	Address	Date of joining	Date of Resigning	Edit	Delete
1	Ramesh	Patel	1968-04-11	xxx yyy zz	2000-11-04	None	Edit	Delete
2	Suresh	Upadhyay	1966-04-19	hdgd	2000-11-08	None	Edit	Delete
3	Jalpa	Asthana	1965-01-24	ttt hhh	2000-09-03	None	Edit	Delete
4	Jaydeep	Parekh	1970-01-05	hhh sjsk	2000-05-04	None	Edit	Delete
5	Feny	Gadhiya	1953-03-25	xxx yyy jjjj	2000-02-07	None	Edit	Delete
6	Shahbaz	Khan	1964-09-01	ddd ff gg	2000-01-08	None	Edit	Delete
7	Falguni	Pathak	1962-11-08	ff shsh sh	2001-08-24	None	Edit	Delete
8	Raj	Malhotra	1958-10-26	yyy ccc xxx	2002-06-29	None	Edit	Delete
9	Abhishek	Patel	1974-01-06	fff rrr sjjs	2000-06-12	None	Edit	Delete
10	Bhakti	Dedania	1977-04-05	vvv xxx zzz	2001-05-18	None	Edit	Delete

We can perform ‘Edit’ command on any tuple by clicking the ‘Edit’ button next to the said tuple. This allows us to edit the various attributes like e_id,e_type, name, gender, phone number, dob, emergency contact, address, date of joining, and date of resigning in existing tuples. All these changes are also reflected in the database in PostgreSQL.



We can perform ‘Delete’ command on any tuple by clicking the ‘Delete’ button next to the said tuple. This allows us to delete the tuple. All these changes are also reflected in the database in PostgreSQL.

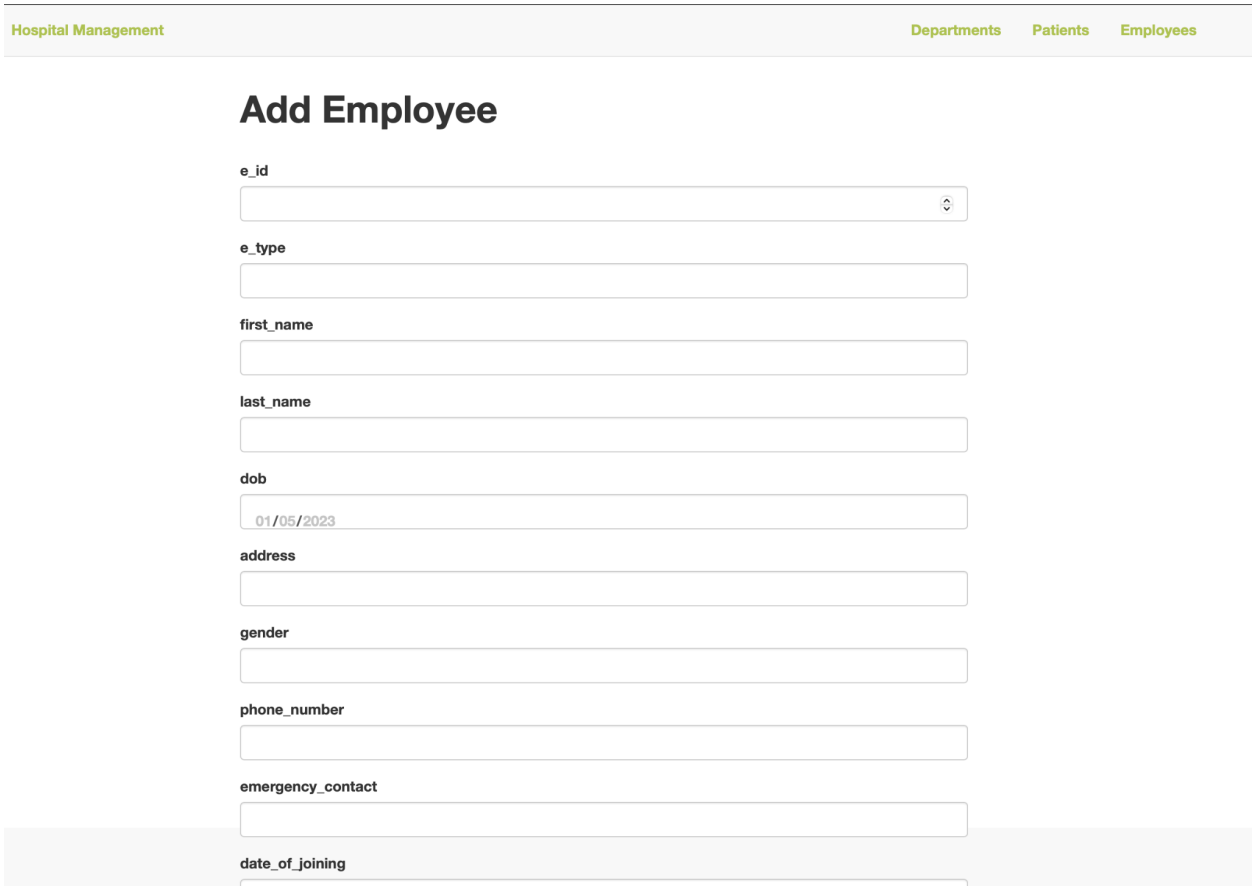


Employee Number	First Name	Last Name	Date of Birth	Address	Date of joining	Date of Resigning	Edit	Delete
1	Ramesh	Patel	1968-04-11	xxx yyy zz	2000-11-04	None	Edit	Delete
2	Suresh	Upadhyay	1966-04-19	hdgd	2000-11-08	None	Edit	Delete
3	Jalpa	Asthana	1965-01-24	ttt hhh	2000-09-03	None	Edit	Delete
4	Jaydeep	Parekh	1970-01-05	hhh sjsk	2000-05-04	None	Edit	Delete
5	Feny	Gadhiya	1953-03-25	xxx yyy jjjj	2000-02-07	None	Edit	Delete
6	Shahbaz	Khan	1964-09-01	ddd ff gg	2000-01-08	None	Edit	Delete
7	Falguni	Pathak	1962-11-08	ff shsh sh	2001-08-24	None	Edit	Delete
8	Raj	Malhotra	1958-10-26	yyy ccc xxx	2002-06-29	None	Edit	Delete
9	Abhishek	Patel	1974-01-06	fff rrr sjjs	2000-06-12	None	Edit	Delete

We can perform ‘Insert’ command on any tuple by clicking ‘Add Employee’ at the end of the relation table. This allows us to add a new tuple to the employee relation. All these changes are also reflected in the database in PostgreSQL.

			04-17		07-10		Edit	Delete
148	Pankti	Dholakia	1979-04-13	jwxbhropv8z6tp61if	2008-03-01	None	Edit	Delete
149	Abhishek	Banerjee	1980-11-11	ij60b937wbuuny7k45	2016-11-07	None	Edit	Delete
150	Dhairya	Shah	1964-07-20	2ddgx40o7qp5xyouoo	2001-02-10	None	Edit	Delete
151	Sujaan	Chatterjee	1972-12-18	ci53upyponzcg07me6	2007-01-01	None	Edit	Delete
152	Omkar	Bhanushali	1951-06-17	djkogu04kqvtxnaynn	2005-07-05	None	Edit	Delete
153	Inder	Vala	1981-08-17	94y2y7g3elrhp5ftlg	2000-01-02	None	Edit	Delete
154	Harin	Popat	1977-03-04	eoj2ur2k0zblh42th7	2005-04-07	None	Edit	Delete
155	Charmi	Ramanuj	1954-05-13	wz81hkal6nknp7uar3	2013-01-11	None	Edit	Delete
156	Gauri	Modi	1977-03-09	lm21ujywoil5wt8t9v	2007-09-11	None	Edit	Delete
157	Faiz	Khan	1978-01-13	ez2u78l23w31yaplvh	2015-01-07	None	Edit	Delete
158	Laksh	Nath	1968-10-20	bdzzewq1gyztsql7hy	2008-02-10	2018-03-05	Edit	Delete
159	Parth	Sharma	1963-04-16	ojgmu5f4w5s9mxdrt2	2008-12-08	None	Edit	Delete

+ Add Employee



9. References:

- **Generation of dummy data - https://www.mockaroo.com/**
- **Avi Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, Seventh Edition, McGraw-Hill, 2019.**
- **Ramakrishnan, R., & Gehrke, J. (2011). *Database Management Systems* (Third). McGraw-Hill.**
- **Garcia-Molina, H., Ullman, J. D., & Widom, J. (2014). *Database systems: The complete book* (Second). Pearson Education Limited.**