

Lab_11

202003018

202003019

Functions

1. prescription of a patient id = x

```
CREATE OR REPLACE FUNCTION prescriptions_patientid(X bigint)
RETURNS SETOF prescription AS $$
DECLARE
    p prescription%rowtype;
BEGIN
    FOR p In Select * FROM prescription WHERE patient_id = X
    Loop
        RETURN NEXT p;
    End Loop;
RETURN;
END;
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM prescriptions_patientid(926722993945);
```

1SELECT * FROM prescriptions_patientid(926722993945)

Data Output

Explain

Messages

Notifications

	patient_id numeric (12)	doctor_id numeric (12)	medicine_id integer	from_date date	to_date date	morning_dose character varying (10)	noon_dose character varying (10)	night_dose character varying (10)	
1	926722993945	131407887378	5	2020-12-04	2020-12-22	7	6	3	
2	926722993945	986786537816	4	2020-12-04	2020-12-10	0	9	7	
3	926722993945	540258634201	8	2020-12-10	2020-12-20	7	3	0	
4	926722993945	131407887378	6	2020-12-04	2020-12-12	7	2	6	

2. lab reports of patient id = x

```
CREATE OR REPLACE FUNCTION reports_patientid(X bigint)
```

```
RETURNS SETOF lab_reports AS $$
```

```
DECLARE
```

```
    r lab_reports%rowtype;
```

```
BEGIN
```

```
    FOR r In SELECT * FROM lab_reports WHERE patient_id = X
```

```
    Loop
```

```
        RETURN NEXT r;
```

```
    End Loop;
```

```
RETURN;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM reports_patientid(983473196869)
```

1

SELECT * FROM reports_patientid(983473196869)

2

Data Output

Explain

Messages

Notifications

	<div>date_time</div> <div>timestamp without time zone</div>	<div>patient_id</div> <div>numeric (12)</div>	<div>type</div> <div>character varying (30)</div>	<div>lab_number</div> <div>smallint</div>
1	2020-11-08 14:09:00	983473196869	MRI	1
2	2020-11-08 09:07:00	983473196869	Blood Test	5

3. prescription of a patient id = x, prescribed by doctor_id = y

```
CREATE OR REPLACE FUNCTION prescription_patientid_doctorid(X  
bigint, Y bigint)
```

```
RETURNS SETOF prescription AS $$
```

```
DECLARE
```

```
    p prescription%rowtype;
```

```
BEGIN
```

```
    For p in select * from prescription where patient_id = X and  
    doctor_id = Y
```

```
    Loop
```

```
        RETURN NEXT p;
```

```
    End Loop;
```

```
RETURN;
```

```
END;
```

```
$$ LANGUAGE plpgsql;
```

```
SELECT * FROM
```

```
prescription_patientid_doctorid(107998063737,596818320348);
```

1SELECT * FROM prescription_patientid_doctorid(107998063737,596818320348)

Data Output

Explain

Messages

Notifications

patient_id numeric (12)	doctor_id numeric (12)	medicine_id integer	from_date date	to_date date	morning_dose character varying (10)	noon_dose character varying (10)	night_dose character varying (10)	
1	107998063737	596818320348	8	2020-09-05	2020-09-25	7	2	3

4. each operation done by doctor with doctorid= X (function)

```
CREATE OR REPLACE FUNCTION operations_doctorid(X bigint)
```

```
RETURNS SET OF operation AS $$
```

```
DECLARE
```

```
o operation%rowtype;
```

```
BEGIN
```

```
For o in SELECT * FROM operation WHERE doctor_id = X
```

```
Loop
```

```
RETURN NEXT o;
```

```
End Loop;
```

```
RETURN;
```

END;

\$\$ LANGUAGE plpgsql;

SELECT * FROM operations_doctorid(540258634201);

1SELECT * FROM operations_doctorid(540258634201)

Data Output

Explain

Messages

Notifications

	patient_id numeric (12)	begin_date_time timestamp without time zone	duration time without time zone	type character varying (20)	doctor_id numeric (12)
1	107998063737	2020-12-04 12:00:00	04:40:00	Lungs	540258634201

5. operation of a patient with patient_id = X

CREATE OR REPLACE FUNCTION operations_patientid(X bigint)

RETURNS SETOF operation AS \$\$

DECLARE

o operation%rowtype;

BEGIN

For o in SELECT * FROM operation WHERE patient_id = X

Loop

RETURN NEXT o;

End Loop;

RETURN;

END;

\$\$ LANGUAGE plpgsql;

SELECT * FROM operation_patientid(713964268158);

1

SELECT * FROM operations_patientid(713964268158)

Data Output

Explain

Messages

Notifications

	<div>patient_id</div> <div>numeric (12)</div>	<div>begin_date_time</div> <div>timestamp without time zone</div>	<div>duration</div> <div>time without time zone</div>	<div>type</div> <div>character varying (20)</div>	<div>doctor_id</div> <div>numeric (12)</div>
1	713964268158	2021-04-07 09:00:00	05:24:00	Cancer	657596761672
2	713964268158	2021-04-07 09:00:00	05:24:00	Cancer	131407887378
3	713964268158	2021-04-07 09:00:00	05:24:00	Cancer	220195004338

6. total unpaid bill of a patient_id = X

CREATE OR REPLACE FUNCTION unpaid_patientid(X bigint)

RETURNS int AS \$\$

DECLARE

unpaid int;

BEGIN



Select Sum(medicine_charges) + Sum(operation_charges) +
Sum(blood_t_charges) + Sum(lab_charges) +
Sum(service_charges) into unpaid From bill where patient_id = X
and status = false;

RETURN unpaid;

END;

\$\$ LANGUAGE plpgsql;

SELECT * FROM unpaid_patientid(618290147720);







1	SELECT * FROM unpaid_patientid(618290147720)		
<div>Data Output</div> <div>Explain</div> <div>Messages</div> <div>Notifications</div>			
	unpaid_patientid integer		
1		34343	

Views

1. available medicines (view for pharmacist)

CREATE VIEW medicines_available AS

SELECT * FROM medicines WHERE amount_available>'0';

Data Output		Explain	Messages	Notifications		
	medicine_id integer	 medicine_name character varying (40)	 cost_per_unit numeric (8,2)	 amount_in_unit smallint	 amount_available integer	 company_name character varying (40)
	1	1 Lithium Carbonate	185.00	10	56	Krajcik Inc
	2	2 Staples Instant Hand Sanitizer	715.00	20	65	Osinski Group
	3	3 Hand Cleanser	236.00	20	43	Heller Group
	4	4 Dexilant	275.00	5	100	Bins-Jacobi
	5	5 Naproxen	460.00	10	49	Leuschke and Sons
	6	6 MICRELL Sp	200.00	12	25	Aufderhar Inc
	7	7 Less Relief	590.00	10	67	Lehner-Thompson
	8	8 Levofloxacin	920.00	5	49	Hayes LLC
	9	9 hyoscyamine sulfate	975.00	5	87	Kreiger-Greenholt
	10	10 METFORMIN HYDROCHLORI...	200.00	10	50	Hauck, Lowe and Steuber

2. bill view (each and every bill) (view for receptionist)

CREATE VIEW bills AS

SELECT * FROM bill ORDER BY date_time DESC;

patient_id numeric (12)	date_time timestamp without time zone	status boolean	medicine_charges numeric (10,2)	blood_t_charges numeric (10,2)	operation_charges numeric (10,2)	lab_charges numeric (10,2)	service_charges numeric (10,2)
1	547192762998 2021-11-09 14:30:00	true	0.00	9065.00	0.00	98528.00	0.00
2	618290147720 2021-10-23 19:10:00	false	0.00	1000.00	0.00	0.00	5500.00
3	618290147720 2021-10-20 20:30:00	false	8504.00	1839.00	0.00	0.00	3000.00
4	760724389956 2021-10-17 11:15:00	false	2340.00	9784.00	0.00	800.00	9000.00
5	618290147720 2021-10-10 19:10:00	false	5000.00	2500.00	0.00	0.00	7000.00
6	618290147720 2021-10-05 19:10:00	true	600.00	500.00	0.00	0.00	7000.00
7	713964268158 2021-08-15 09:00:00	true	10000.00	2500.00	125000.00	8700.00	70036.00
8	578283562069 2021-06-29 18:30:00	true	2540.00	6648.00	0.00	0.00	9700.00
9	926722993945 2021-01-09 10:30:00	true	9000.00	4069.00	0.00	2400.00	6300.00
10	923692306899 2021-01-04 12:00:00	true	5000.00	4445.00	0.00	2500.00	0.00
11	983473196869 2020-09-09 08:30:00	true	4860.00	187.00	0.00	0.00	0.00
12	107998063737 2020-09-05 15:30:00	true	8000.00	3352.00	0.00	0.00	0.00
13	707819011547 2020-03-03 13:30:00	true	6000.00	6859.00	0.00	0.00	0.00

3. can get patient list for every room (view for nurses)

CREATE VIEW rooms_patientids AS


```
SELECT * FROM admitted_patients_ids ORDER BY room_no,
date_of_admit DESC;
```

Data Output	Explain	Messages	Notifications
	room_no integer	date_of_admit date	patient_id numeric (12)
1	1	2020-12-04	926722993945
2	2	2021-07-08	713964268158
3	3	2021-06-01	578283562069
4	5	2021-10-06	760724389956

4. medicine view (view for pharmacist)

```
CREATE VIEW medicine_detail AS
```

```
SELECT * FROM medicines ORDER BY medicine_id;
```

medicine_id integer	medicine_name character varying (40)	cost_per_unit numeric (8,2)	amount_in_unit smallint	amount_available integer	company_name character varying (40)
1	Lithium Carbonate	185.00	10	56	Krajcik Inc
2	Staples Instant Hand Sanitizer	715.00	20	65	Osinski Group
3	Hand Cleanser	236.00	20	43	Heller Group
4	Dexilant	275.00	5	100	Bins-Jacobi
5	Naproxen	460.00	10	49	Leuschke and Sons
6	MICRELL Sp	200.00	12	25	Aufderhar Inc
7	Less Relief	590.00	10	67	Lehner-Thompson
8	Levofloxacin	920.00	5	49	Hayes LLC
9	hyoscyamine sulfate	975.00	5	87	Kreiger-Greenholt
10	METFORMIN HYDROCHLORI...	200.00	10	50	Hauck, Lowe and Steuber

5. prescription view (excluding doctor and patient details)(view for pharmacist)

CREATE VIEW prescriptions_medicines AS

SELECT medicine_id, from_date, to_date, morning_dose, noon_dose, night_dose FROM prescription ORDER BY medicine_id, to_date DESC;

	Data Output	Explain	Messages	Notifications					
	medicine_id Integer		from_date date	to_date date	morning_dose character varying (10)	noon_dose character varying (10)	night_dose character varying (10)		
1	1		2020-03-03	2020-03-12	7	9	4		
2	2		2021-06-10	2021-06-20	3	3	8		
3	4		2020-12-04	2020-12-10	0	9	7		
4	5		2020-12-04	2020-12-22	7	6	3		
5	6		2020-12-04	2020-12-12	7	2	6		
6	6		2020-09-05	2020-09-12	8	6	7		
7	7		2021-01-04	2021-01-27	0	1	5		
8	8		2020-12-10	2020-12-20	7	3	0		
9	8		2020-09-05	2020-09-25	7	2	3		
10	9		2021-07-10	2021-07-20	5	1	4		

6. can get patient list for every room (view for nurses)

CREATE VIEW rooms_patientids AS

SELECT * FROM admitted_patients_ids ORDER BY room_no, date_of_admit DESC;

Data Output Explain Messages Notifications

	 room_no integer 	date_of_admit date 	patient_id numeric (12) 	
1	1	2020-12-04	926722993945	
2	2	2021-07-08	713964268158	
3	3	2021-06-01	578283562069	
4	5	2021-10-06	760724389956	