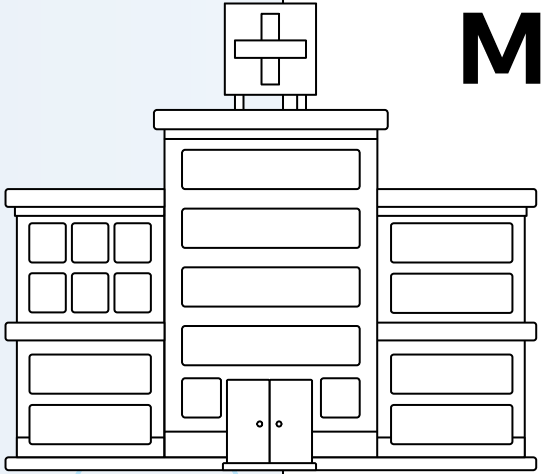
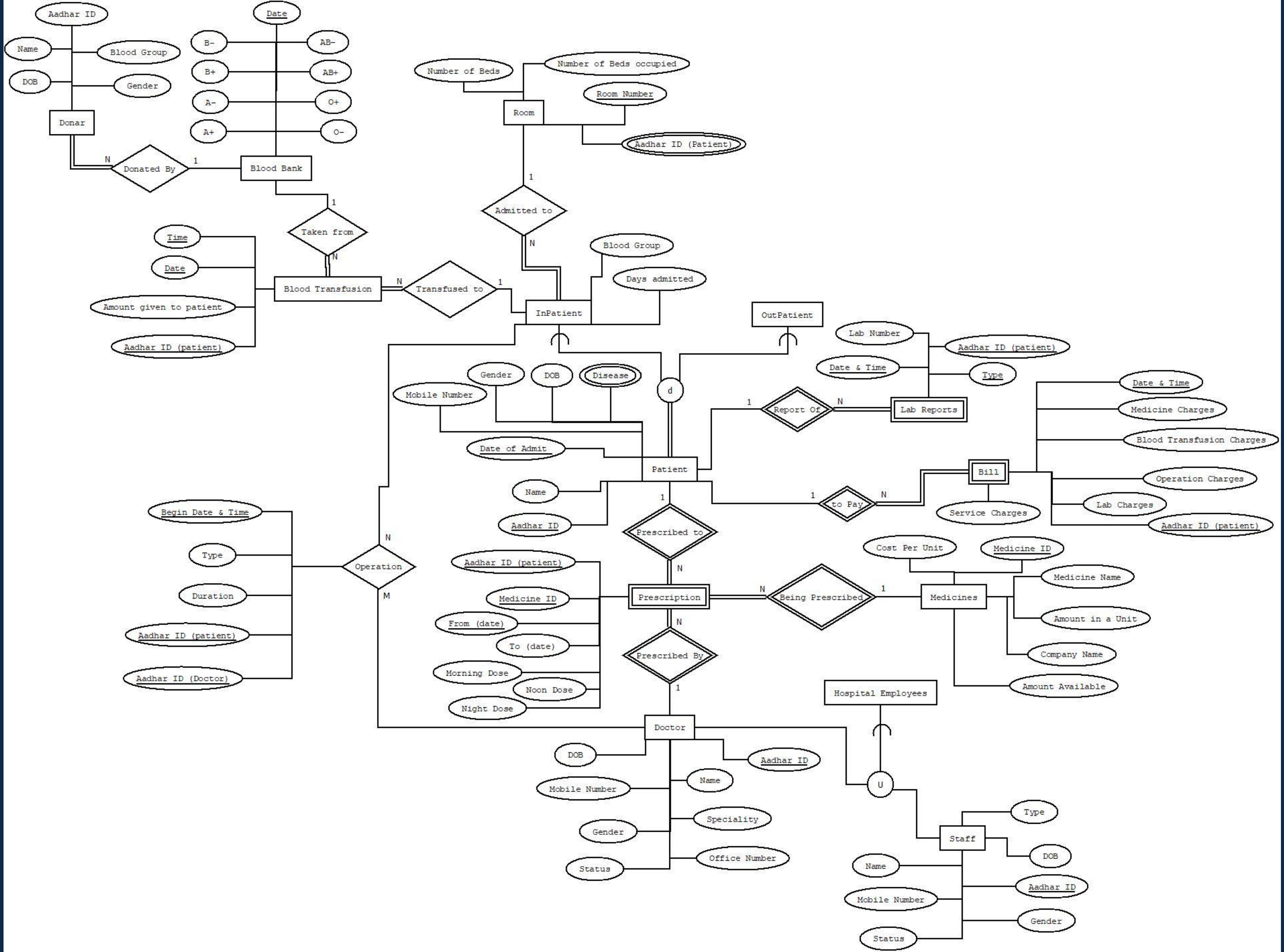
Hospital Database Management System

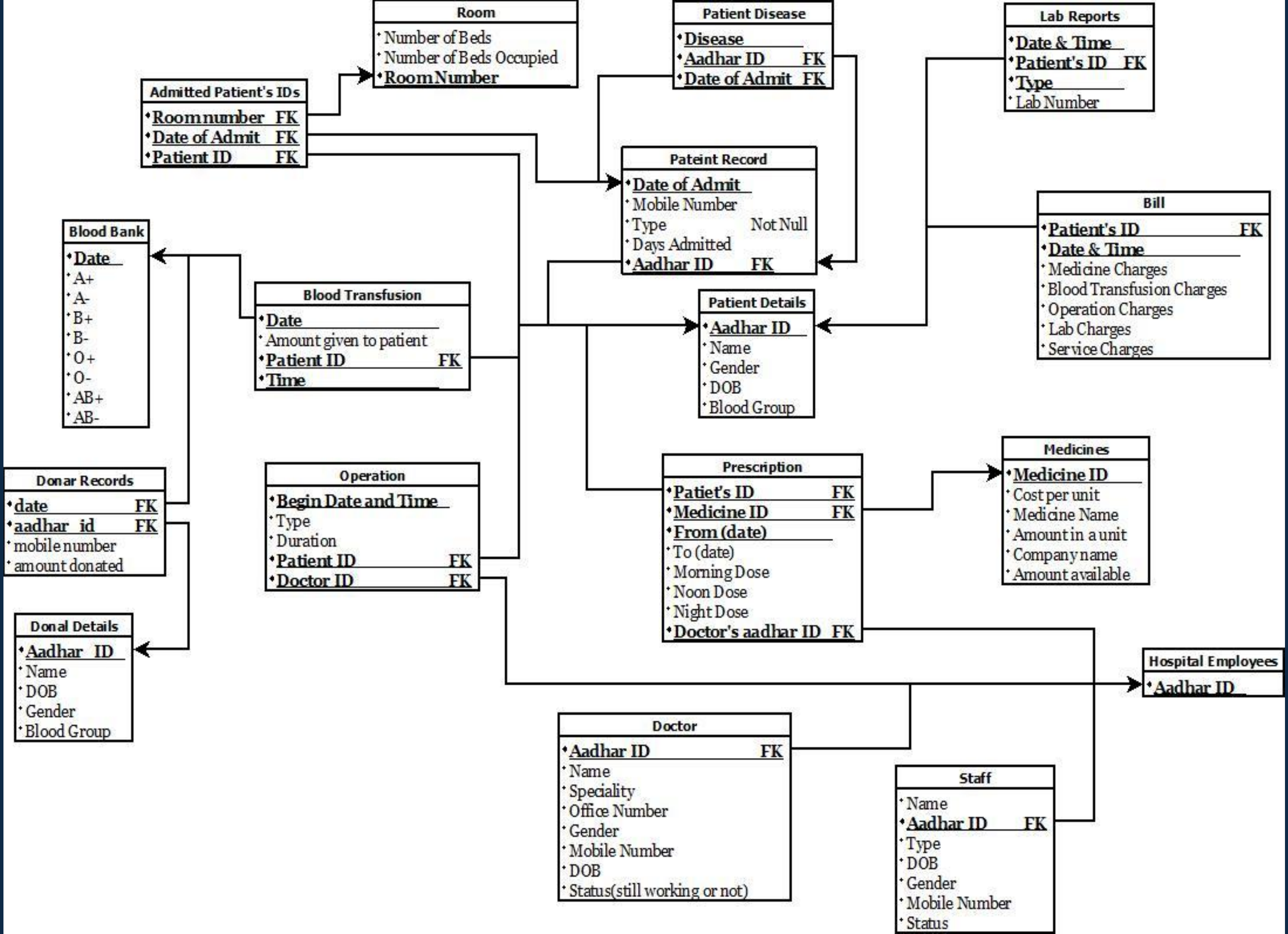


Kanishk Menaria	202003018
Jay Patel	202003019

Functional Requirements

1. There will be two types of patients: in-patient and out-patient, in patient will be assigned a room in hospital.
2. Each entry of patient will be recorded in the database.
3. Hospital may have many patients identified with their unique id.
4. Prescriptions mentioned by the doctors to the patients will be saved.
5. Blood Bank entity will store the data of amount of blood preserved in the blood bank of hospital.
6. Database will also contain data of donors who have donated blood and amount of blood.
7. Each entity of type Blood Transfusion will be connected to a single entity of Blood Bank entity.
8. Data of laboratory checkup will be stored in the database. It will contain details such as patient id, type of test, lab number and Date&Time.
9. Bill will contain patient id, Date&Time, laboratory charge, medicine charges, operation charge and other charges.
10. The data of medicines which are available at the hospital will be saved in the database.
11. More than one patient can be admitted to one room.
12. If any operation happens in the hospital, then the details of the operation such as doctors evolved in it, patient, operation type, and its timing will be saved in the database.
13. One patient can be treated by one or more than one doctor during an operation.
14. All the data will be updated regularly by the operational staff.


Aadhar ID
Name
Blood Group
DOB
Gender
Donar
Donated By
N
1
Date
B-
AB-
B+
AB+
A-
O+
A+
O-
Blood Bank
1
Taken from
N
Time
Date
Amount given to patient
Aadhar ID (patient)
Blood Transfusion
N
Transfused to
1
Number of Beds
Number of Beds occupied
Room Number
Room
Aadhar ID (Patient)
1
Admitted to
N
Blood Group
Days admitted
InPatient
OutPatient
d
Gender
DOB
Disease
Mobile Number
Date of Admit
Name
Aadhar ID
Patient
Lab Number
Aadhar ID (patient)
Date & Time
Type
1
Report Of
N
Lab Reports
Date & Time
Medicine Charges
Blood Transfusion Charges
Bill
Operation Charges
Lab Charges
Service Charges
Aadhar ID (patient)
1
to Pay
N
1
Prescribed to
N
Begin Date & Time
Type
Duration
Aadhar ID (patient)
Aadhar ID (Doctor)
N
Operation
M
Aadhar ID (patient)
Medicine ID
From (date)
To (date)
Morning Dose
Noon Dose
Night Dose
Prescription
N
Being Prescribed
1
Cost Per Unit
Medicine ID
Medicine Name
Medicines
Amount in a Unit
Company Name
Amount Available
N
Prescribed By
1
Doctor
DOB
Aadhar ID
Mobile Number
Name
Speciality
Gender
Office Number
Status
Hospital Employees
U
Staff
Type
DOB
Name
Aadhar ID
Mobile Number
Gender
Status

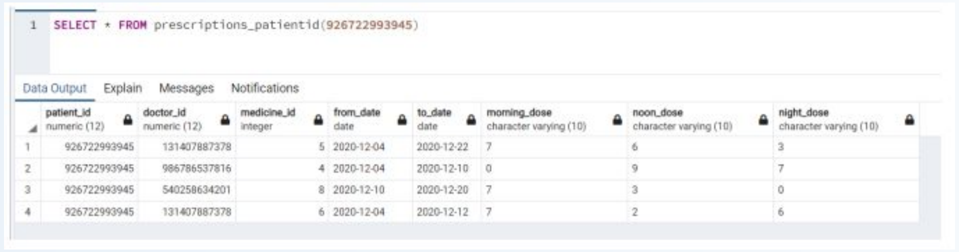

Room
Number of Beds
Number of Beds Occupied
Room Number
Patient Disease
Disease
Aadhar ID FK
Date of Admit FK
Lab Reports
Date & Time
Patient's ID FK
Type
Lab Number
Admitted Patient's IDs
Room number FK
Date of Admit FK
Patient ID FK
Pateint Record
Date of Admit
Mobile Number
Type Not Null
Days Admitted
Aadhar ID FK
Bill
Patient's ID FK
Date & Time
Medicine Charges
Blood Transfusion Charges
Operation Charges
Lab Charges
Service Charges
Blood Bank
Date
A+
A-
B+
B-
O+
O-
AB+
AB-
Blood Transfusion
Date
Amount given to patient
Patient ID FK
Time
Patient Details
Aadhar ID
Name
Gender
DOB
Blood Group
Donar Records
date FK
aadhar id FK
mobile number
amount donated
Operation
Begin Date and Time
Type
Duration
Patient ID FK
Doctor ID FK
Prescription
Patiet's ID FK
Medicine ID FK
From (date)
To (date)
Morning Dose
Noon Dose
Night Dose
Doctor's aadhar ID FK
Medicines
Medicine ID
Cost per unit
Medicine Name
Amount in a unit
Company name
Amount available
Donal Details
Aadhar ID
Name
DOB
Gender
Blood Group
Hospital Employees
Aadhar ID
Doctor
Aadhar ID FK
Name
Speciality
Office Number
Gender
Mobile Number
DOB
Status(still working or not)
Staff
Name
Aadhar ID FK
Type
DOB
Gender
Mobile Number
Status

SQL DDL Statements

```
CREATE SCHEMA hospital;

SET SEARCH_PATH TO hospital;

CREATE TABLE hospital_employees (

.

.

FOREIGN KEY (patient_id) REFERENCES patient_details(aadhar_id)

ON DELETE RESTRICT ON UPDATE CASCADE );
```

https://daiictacin-my.sharepoint.com/:w:/g/personal/202003019_daiict_ac_in/EWKrX7UV0u5Er2Uuj_-HxUYBMfNu4_vxqHqnSy_wYHMCgA?e=CxA0Zy

Prescription of a patient id = x (function)



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

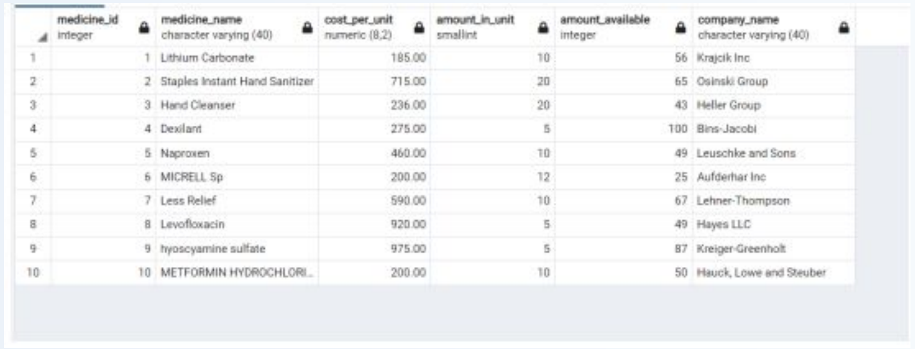
Data Output Explain Messages Notifications

	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

Here, X = 926722993945

```
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;
```

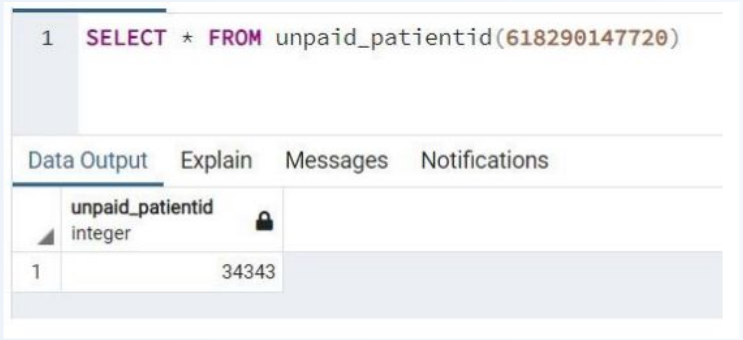
Medicine view (view for pharmacist)



	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

```
CREATE VIEW medicine_detail AS

SELECT * FROM medicines ORDER BY medicine_id;
```

Total unpaid bill of a patient_id = X (function)



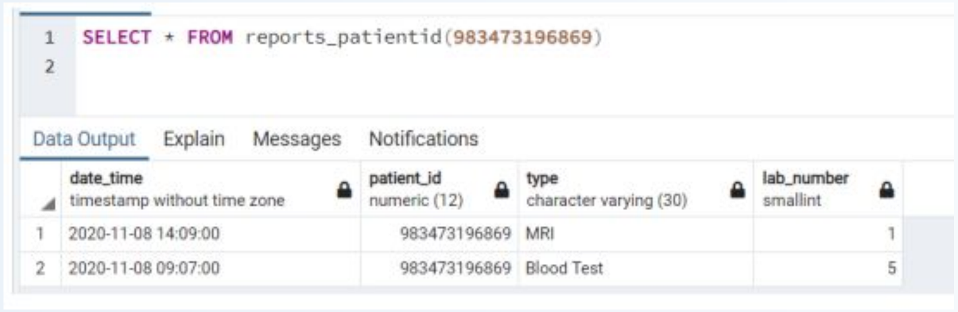
```
SELECT * FROM unpaid_patientid(618290147720)
```

```
   unpaid_patientid
   integer
1             34343
```

Here, X = 618290147720

```
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;
```

Lab reports of patient id = x (function)



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

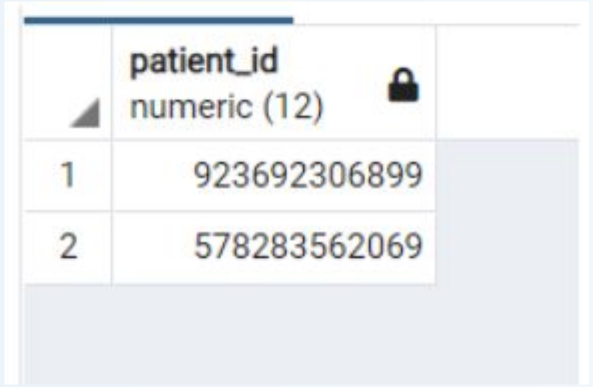
Data Output | Explain | Messages | Notifications

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

Here, X = 983473196869

```
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;
```

One can get list of patients with similar disease = ‘X’ (queries)

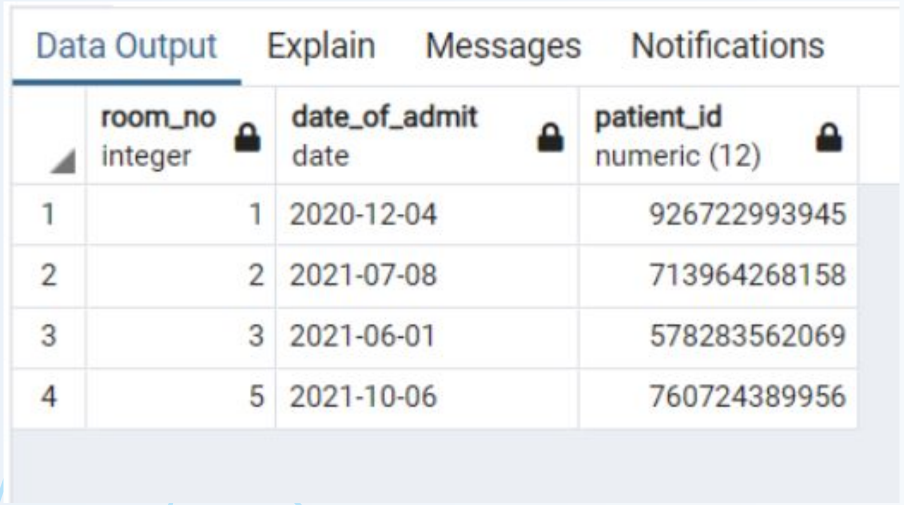


	patient_id numeric (12)
1	923692306899
2	578283562069

Here, X = ‘DENGUE'

Select patient_id from patient_disease where upper(disease) = ‘X’

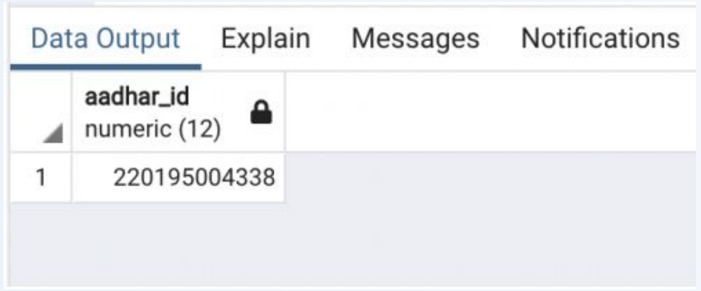
Can get patient list for every room (view for nurses)


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

```
CREATE VIEW rooms_patientids AS

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

We get details of doctors which were present in every operation of a patient id = X (queries)



```
Data Output   Explain   Messages   Notifications

     aadhar_id
     numeric (12)
1    220195004338
```

Here, X = 713964268158

```
Select aadhar_id from doctor

except

(Select id from

(Select doctor.aadhar_id as id, O.begin_date_time
from doctor cross join (select * from operation
where patient_id = X) as O

except

Select doctor_id, begin_date_time from operation
where patient_id = X ) as D );
```

Java code with JDBC API

```
package lab12;

import java.sql.Connection;

import java.sql.DriverManager;

.

.

System.exit(0); }

}

}
```

https://daiictacin-my.sharepoint.com/:w:/g/personal/202003019_daiict_ac_in/EXDoFZQm6sxEs_xM1_Z3Yf4BdXpDCoakdGJszQDc7Z-rVw?e=K8acfI

Conclusion

How our project is helpful in real life ?

Hospital database systems play an important role in real life because they store vital information about patients, doctors, and all activities that occur or have occurred in the hospital. This aids in obtaining hospital-related data in a matter of seconds.

What did we learn ?

While working on this project, we learnt database related concepts such as creating Entity Relation Diagram, creating Relational Schema, writing DDL statements, finding Functional Dependencies, etc. Except that we also gained a better understanding of how a hospital operates.