

DBMS PROJECT

**Teacher: Debanjan Sadhya
Sir**





PRESENTED BY:

ANIL KUMAR 2020BCS-011

DEEPANSHU JAIN 2020BCS-025

G. NAGENDRA 2020BCS-035

NEETU KUMARI 2020BCS-052

SHUBHAM GANDHI 2020BCS-070

SUKHVEER SINGH 2020BCS-071

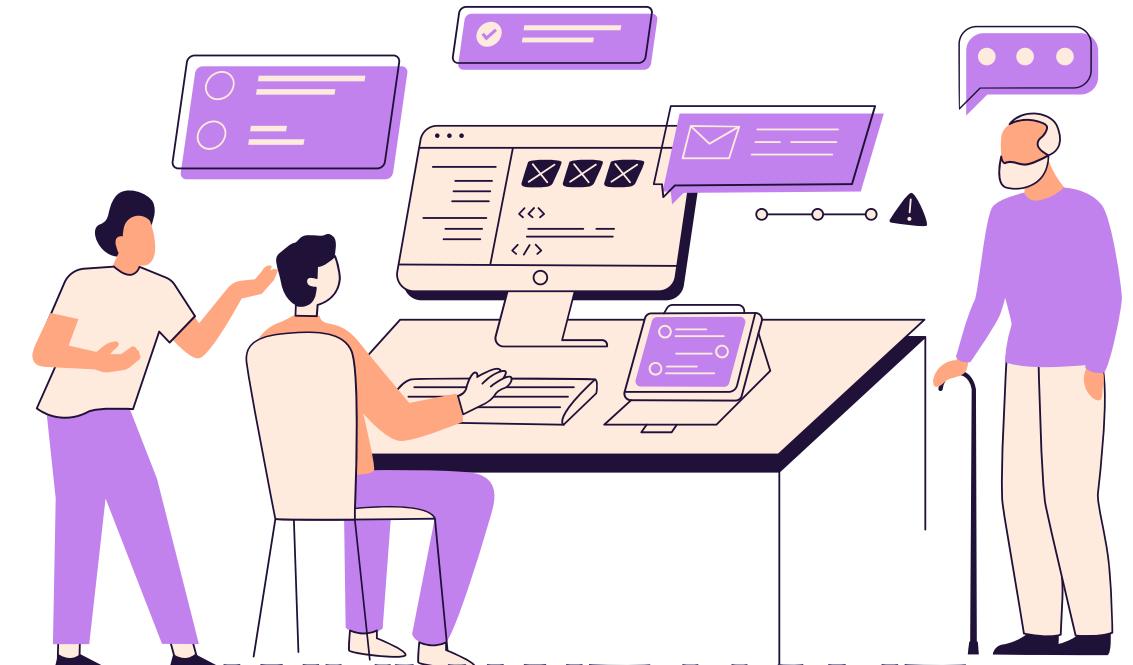
TOPIC: HOSPITAL MANAGEMENT SYSTEM

PROJECT DESCRIPTION

THIS PROJECT IS ABOUT CREATING AND MANAGING A HOSPITAL DATABASE SYSTEM. THIS PROJECT OFFER USER TO STORE PATIENT, DOCTOR, NURSE, PATHOLOGY, BILLING, PHARMACEUTICAL STORE, ALL IN ONE DATABASE.

ADMINISTER CAN VIEW FOR THE PATIENTS: WHEN, WHY, WHERE THEY ADMITTED.

THIS DATABASE ALSO HELPS TO STORE THE INFORMATION REGARDING EMPLOYEE INFORMATION LIKE DATE OF BIRTH, CONTACT NO, ADDRESS, ETC. THIS DATABASE IS BASED ON MANY ASSUMPTIONS AND EXPECTATIONS ARE ALSO TAKEN INTO CONSIDERATION AS THE REAL-LIFE DATABASE WOULD BE EXTREMELY LARGE AND IT WOULD NOT BE POSSIBLE TO TAKE A CASE STUDY ON SUCH A LARGE DATABASE.



WHY WE HAVE CHOOSE THIS TOPIC

HOSPTIALS ARE THE ESSENTIAL PART OF OUR LIVES, PROVIDING BEST MEDICAL FACILITIES TO PEOPLE SUFFERING FROM VARIOUS ALIMENTS, WHICH MAY BE DUE TO CHANGE IN CLIMATIC CONDITIONS, INCREASE WORK-LOAD, EMOTIONAL TRAUMA, STRESS ,ETC. IT IS NECESSARY FOR THE HOSPTIAL TO KEEP OF TRACK OF ITS DAY-TO-DAY ACTIVITES & RECORDS OF ITS PATIENTS, DOCTOR, NURSE AND OTHER IMFROMATION REGARDING HOSPTIAL TO KEEP IT RUNNING SMOOTH & SUCCESSFULL

ENTITY SETS

- DOCTOR
- PATIENT
- NURSE
- BILL
- ROOM
- EMPLOYEE
- DEPARTMENT
- TEST
- OUT_PATIENT
- MEDICINE

RELATIONSHIP SETS

TAKES

MANY TO MANY FROM PATIENT TO MEDICINE ENTITY SET AS ONE PATIENT CAN TAKE MANY MEDICINES AND ONE MEDICINE CAN BE TAKEN BY MANY PATIENTS

DOCTOR_ASSIGNED

MANY TO MANY RELATION FROM DOCTOR TO PATIENTS ENTITY SET AS ONE PATIENT CAN BE CHECKED BY MULTIPLE DOCTOR AND ONE DOCTOR CAN BE ASSIGNED TO MULTIPLE PATIENTS.

HAS

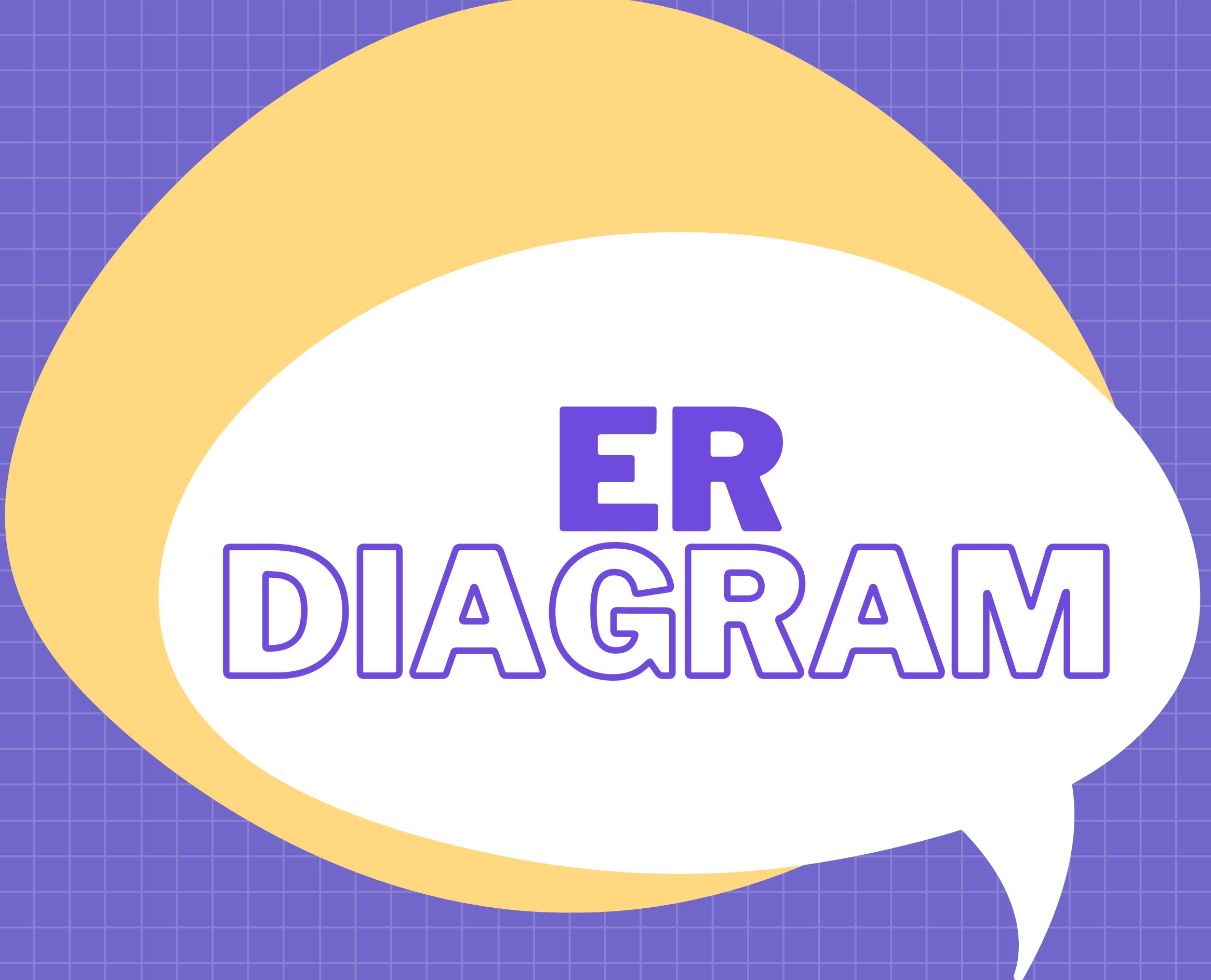
MANY TO MANY FROM PATIENT TO TEST ENTITY SET AS ONE PATIENT CAN HAVE MANY TESTS AND TEST CAN ALSO BE TAKEN BY MANY PATIENTS.

IN_PATIENT

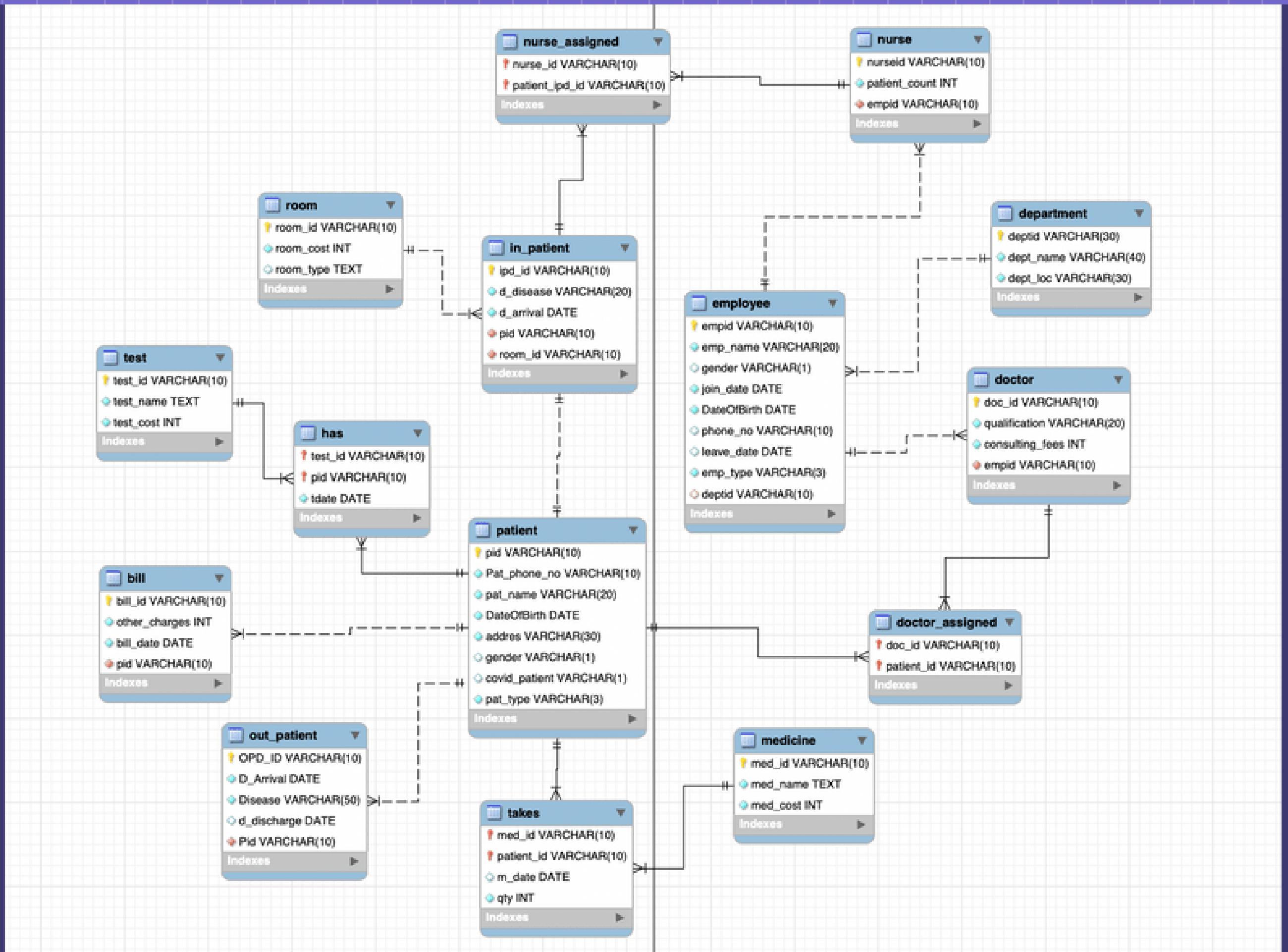
ONE TO MANY FROM ROOM TO PATIENT AS A ROOM CAN HAVE MANY PATIENTS BUT ONE PATIENT CAN'T HAVE MORE THAN ONE ROOMS.

NURSE_ASSIGNED

MANY TO MANY FROM PATIENTS TO NURSE ENTITY SET AS ONE PATIENT CAN BE CHECKED BY MULTIPLE NURSE, AND A NURSE CAN CHECK MULTIPLE PATIENTS.



ER DIAGRAM



INITIAL SCHEMAS

- Doctor(doc_id,qualification,consulting_fees, empid)
- Nurse (nurseid,patient_count,empid)
- Patient(Pid,Pat_phone_no,pat_name,DateOfBirth,address,gender,covid_patient ,pat_type)
- Department(deptid,dept_name,dept_loc)
- Medicine(med_id,med_name,med_cost)
- Test(test_id,test_name,test_cost)
- Bill(bill_id,other_charges,bill_date,pid)
- Out_patient(OPD_ID,D_Arrival,Disease,d_discharge,Pid)
- Room(room_id,room_cost,room_type)
- Employee(empid, emp_name,gender,join_date, DateOfBirth,phone_no ,leave_date,emp_type,deptid)

FUNCTIONAL DEPENDENCIES

- Doctor
 - $F = \{doc_id \rightarrow (qualification, consulting_fees, empid),$
 $qualification \rightarrow consulting_fees\}$
- Nurse
 - $F = \{nurseid \rightarrow (patient_count, empid)\}$
- Patient
 - $F = \{Pid \rightarrow (Pat_phone_no, pat_name, DateOfBirth, address, gender,$
 $covid_patient, pat_type),$
 $covid_patient \rightarrow pat_type\}$
- Department
 - $F = \{deptid \rightarrow (dept_name, dept_loc)\}$

- Test
 $F = \{test_id \rightarrow (test_name, test_cost)\}$
- Bill.
 $F = \{bill_id \rightarrow (other_charges, bill_date, pid)\}$
- Out_patient
 $F = \{OPD_ID \rightarrow (D_Arrival, Disease, d_discharge, Pid)\}$
- Room
 $F = \{room_id \rightarrow (room_cost, room_type),$
 $room_type \rightarrow room_cost\}$
- Employee
 $F = \{empid \rightarrow (emp_name, gender, join_date, DateOfBirth, phone_no,$
 $leave_date, emp_type, deptid)\}$
- Medicine
 $F = \{med_id \rightarrow (med_name, med_cost)\}$

1ST NORMAL FORM

Assuming all the attributes of all relations are already atomic, we can infer that all relations are already in 1NF.

2ND NORMAL FORM

Conditions for 2nd normal form are-

- It should be in 1NF
- It should not have any partial dependency.

So, the table is in 1st normal form and now for partial dependency, there is no partial dependency as all entity sets have only one primary key so there is no chance of partial dependency.

Hence all relations are in 2NF.

3RD NORMAL FORM

- For 3rd normal form –
 - It should be in 1st and 2nd NF
 - It should not have a transitive dependency
- So there exist transitive in Doctor, Patient and room entity sets

If $P \rightarrow Q$ and $Q \rightarrow R$ is true, then $P \rightarrow R$ is a transitive dependency.

In Doctor set $\text{doc_id} \rightarrow (\text{qualification}, \text{consulting_fees}, \text{empid})$, $\text{qualification} \rightarrow \text{consulting_fees}$
So in this transitive dependency occurs so, we decompose it to two on the basis of its closure.

In Patient set $\text{Pid} \rightarrow (\text{Pat_phone_no}, \text{pat_name}, \text{DateOfBirth}, \text{address}, \text{gender}, \text{covid_patient}, \text{pat_type})$, $\text{covid_patient} \rightarrow \text{pat_type}$, So in this also transitive dependency occurs so decompose it to two on the basis of its closure.

In Room set $\text{room_id} \rightarrow (\text{room_cost}, \text{room_type})$, $\text{room_type} \rightarrow \text{room_cost}$
So in this also transitive dependency occurs so decompose it to two on the basis of its closure.



Schemas after decomposing relations to 3rd NF-

- Doctor(doc_id , qualification, empid)
- Doc_qual(qualification,consulting_fees)
- Nurse (nurseid,patient_count,empid)
- Patient(Pid,Pat_phone_no,pat_name,DateOfBirth,address,gender,covid_patient)
 - Cov_patient(covid_patient,pat_type)
- Department(deptid,dept_name,dept_loc)
- Medicine(med_id,med_name,med_cost)
- Test(test_id,test_name,test_cost)
- Bill(bill_id,other_charges,bill_date,pid)
- Out_patient(OPD_ID,D_Arrival,Disease,d_discharge,Pid)
 - Room(room_id,room_type)
 - Type_cost(room_cost,room_type)
- Employee(empid, emp_name,gender,join_date, DateOfBirth,phone_no , leave_date, emp_type,deptid)

FUNCTIONAL DEPENDENCIES

- Doctor
 $F = \{doc_id \rightarrow empid, qualification\}$
- Doc_qual
 $F = \{qualification \rightarrow consulting_fees\}$
- Nurse
 $F = \{nurseid \rightarrow (patient_count, empid)\}$
- Patient
 $F = \{Pid \rightarrow (Pat_phone_no, pat_name, DateOfBirth, address, gender, covid_patient)\}$
- Cov_patient
 $F = \{covid_patient \rightarrow pat_type\}$
- Department
 $F = \{deptid \rightarrow (dept_name, dept_loc)\}$

- Medicine
 $F = \{med_id \rightarrow (med_name, med_cost)\}$
- Test
 $F = \{test_id \rightarrow (test_name, test_cost)\}$
- Bill
 $F = \{bill_id \rightarrow (other_charges, bill_date, pid)\}$
- Out_patient
 $F = \{OPD_ID \rightarrow (D_Arrival, Disease, d_discharge, Pid)\}$
- Room
 $F = \{room_id \rightarrow room_type\}$
- Type_cost
 $F = \{room_type \rightarrow room_cost\}$
- Employee
 $F = \{empid \rightarrow (emp_name, gender, join_date, DateOfBirth, phone_no, leave_date, emp_type, deptid)\}$

BCNF NORMAL FORM

From the functional dependency (formed after 3rd normal form) We get to know that in all functional dependencies LHS of them is a candidate key in their respective entity set

All entity sets are reduced to BCNF

ALL THE TABLES AFTER REDUCING UPTO BCNF

1. DEPARTMENT

deptid	dept_name	dept_loc
DPT01	Physiotherapy	Building 3, left wing A block
DPT02	Neurology	Building 2, left wing C block
DPT03	Ortho	Building 3, left wing A block
DPT04	Psychology	Building 1, left wing B block
DPT05	Gynaecology	Building 1, left wing C block
DPT06	Medicine	Building 1, left wing A block
DPT07	Gastroentrology	Building 3, left wing B block
DPT08	Cardiac	Building 2, left wing A block
DPT09	ENT	Building 3, left wing A block
DPT10	General Surgery	Building 2, left wing B block
DPT11	Anaesthetics	Building 1, left wing A block
NULL	NULL	NULL

2. DOCTOR

doc_id	qualification	empid
DOC01	MBBS	EM001
DOC02	MBBS	EM002
DOC03	MBBS	EM003
DOC04	MD	EM004
DOC05	MS	EM005
DOC06	MD	EM006
DOC07	MS	EM007
DOC08	MD	EM008
DOC09	MD	EM009
NULL	NULL	NULL

3. NURSE

nurseid	patient_count	empid
NUR01	1	EM010
NUR02	1	EM011
NUR03	1	EM012
NUR04	2	EM013
NUR05	1	EM014
NUR06	2	EM015
NUL	NUL	NUL

4. PATIENT

5. EMPLOYEE

6.OUT_PATIENT

	OPD_ID	D_Arrival	Disease	d_discharge	Pid
▶	OPD01	2020-10-21	Heart attack	2020-12-14	PAT001
	OPD02	2021-04-13	Gastric	2021-05-14	PAT004
	OPD03	2021-04-16	Lung Infection	2021-05-30	PAT006
	OPD04	2020-09-27	Ear Pain	2020-10-02	PAT007
	OPD05	2021-01-21	Fever	2021-01-25	PAT009
	OPD06	2021-03-13	Throat Infection	2021-04-11	PAT011
	OPD07	2021-05-09	Lung Infection	2021-06-07	PAT012
	OPD08	2021-04-12	Covid	2021-04-22	PAT013
	OPD09	2021-05-01	White Fungus	2021-05-22	PAT014
	NULL	NULL	NULL	NULL	NULL

7.IN_PATIENT

	ipd_id	d_disease	d_arrival	pid	room_id
	IPD01	Heart-Attack	2021-06-19	PAT002	R001
	IPD02	Gestational-Diabetes	2021-06-17	PAT003	R002
	IPD03	Kidney Transplant	2021-06-04	PAT005	R004
	IPD04	Hernia	2021-05-25	PAT008	R005
	IPD05	Femur Fracture	2021-06-12	PAT010	R005
	IPD06	Appendicitis	2021-05-30	PAT015	R007
	NULL	NULL	NULL	NULL	NULL

8. ROOM

room_id	room_type
R001	single
R002	double
R003	dorm
R004	single
R005	double
R006	dorm
R007	single
R008	double
R009	dorm
R010	single
R011	double
R012	dorm
NULL	NULL

9. MEDICINE

med_id	med_name	med_cost
MED01	Remdesivir	900
MED02	Citzin	70
MED03	Aspirin	1000
MED04	Wararin	2000
MED05	Insulin-A	600
MED06	Glynase	700
MED07	Paracetamol	100
NULL	NULL	NULL

10.TEST

test_id	test_name	test_cost
► TES01	Troponin Test	1500
TES02	CK-MB Test	2000
TES03	Serum Myoglobin Test	1700
TES04	Random Blood Sugar Test	500
TES05	Fasting Blood Sugar Test	500
TES06	IC PCR Test	1000
HULL	HULL	HULL

11.TAKES

med_id	patient_id	m_date	qty
MED01	PAT001	2020-11-21	2
MED01	PAT002	2021-06-24	3
MED01	PAT013	2021-04-18	10
MED01	PAT014	2020-05-03	10
MED02	PAT004	2021-04-20	8
MED02	PAT009	2011-05-26	10
MED04	PAT005	2021-06-27	2
MED04	PAT010	2021-06-23	10
MED04	PAT012	2021-06-02	10
MED05	PAT003	2021-06-22	6
MED05	PAT007	2021-09-28	10
MED05	PAT011	2021-03-14	10
MED05	PAT015	2020-06-06	10
MED06	PAT006	2021-04-19	10
MED07	PAT008	2021-01-22	10
HULL	HULL	HULL	HULL

12.BILL

bill_id	other_charges	bill_date	pid
BILL01	1000	2020-12-14	PAT001
BILL02	100000	2021-06-30	PAT002
BILL03	900	2021-06-22	PAT003
BILL04	1200	2021-05-14	PAT004
BILL05	55000	2021-06-22	PAT005
BILL06	2000	2021-05-30	PAT006
BILL07	1600	2020-10-02	PAT007
BILL08	35000	2021-06-10	PAT008
BILL09	1100	2021-01-25	PAT009
► BILL10	65000	2021-06-30	PAT010
BILL11	3000	2021-04-11	PAT011
BILL12	1200	2021-06-07	PAT012
BILL13	1000	2021-04-22	PAT013
BILL14	900	2021-05-22	PAT014
BILL15	90000	2021-06-21	PAT015
NULL	NULL	NULL	NULL

13.HAS

test_id	pid	tdate
TES02	PAT002	2021-06-20
TES03	PAT004	2021-05-09
TES04	PAT008	2021-05-30
TES05	PAT015	2021-06-01
► TES06	PAT002	2021-06-20
► TES06	PAT003	2021-06-19
TES06	PAT010	2021-06-16
NULL	NULL	NULL

14.COV_PATIENT

covid_patient	pat_type
n	opd
y	ipd
NULL	NULL

15.TYPE_COST

room_cost	room_type
1000	dorm
2000	double
4000	single
NULL	NULL

16.DOC_QUAL

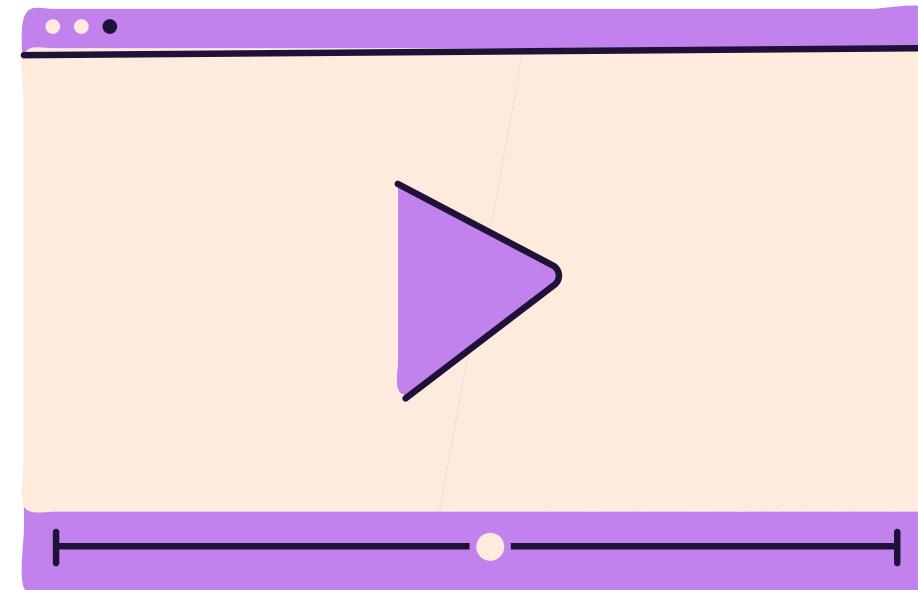
qualification	consulting_fees
MBBS	3000
MD	8000
MS	4000
NULL	NULL



SQL
QUERY

QUESTION 1

Write sql query that lists all the number of male covid patients.



QUERY 1

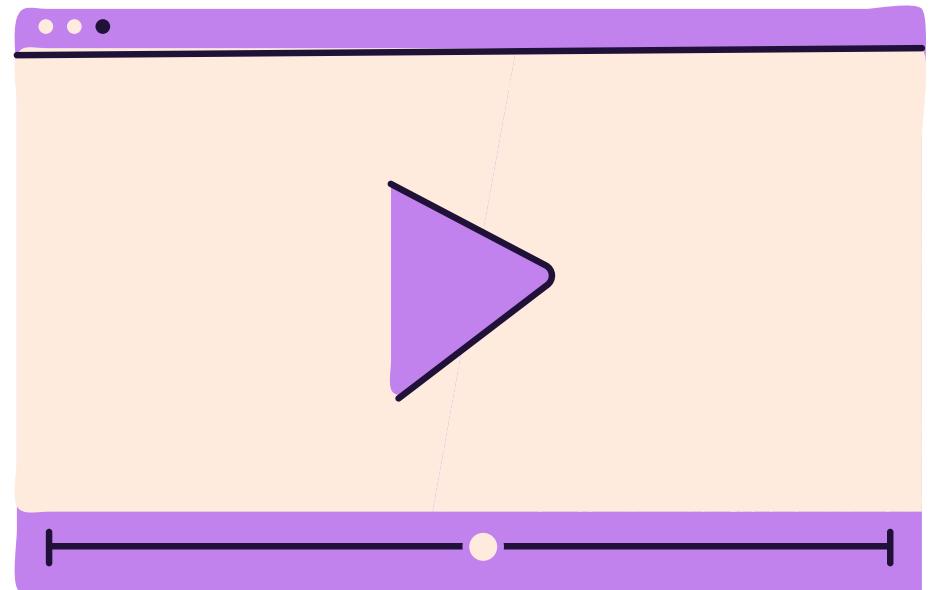
```
select  
pid,pat_name,pat_phone_no from  
patient where gender = 'm' and  
covid_patient = 'y';
```

OUTPUT

pid	pat_name	pat_phone_no
PAT010	Tushar	9123456789
PAT011	Karan	0223456789
PAT013	Harsh	0423456789
PAT015	Deepak	0623456789
NULL	NULL	NULL

QUESTION 2

Write a SQL query to calculate the total sum of all the other charges made by all patient.



QUERY 2

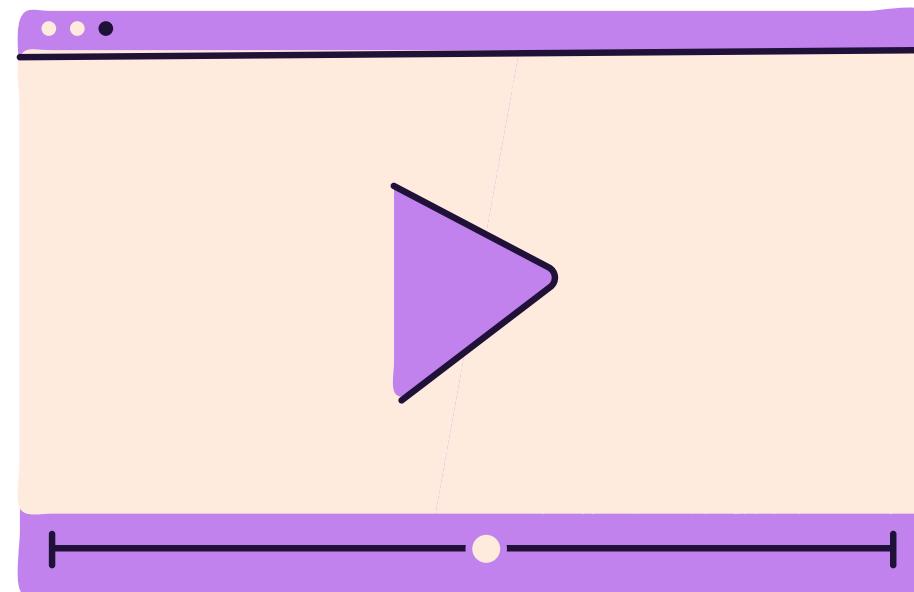
```
select sum(other_charges) as  
total_charges from bill ;
```

OUTPUT

total_charges
▶ 358900

QUESTION 3

Write a SQL query that lists name
employee id and date of birth of all
female nurses born after the year
1990



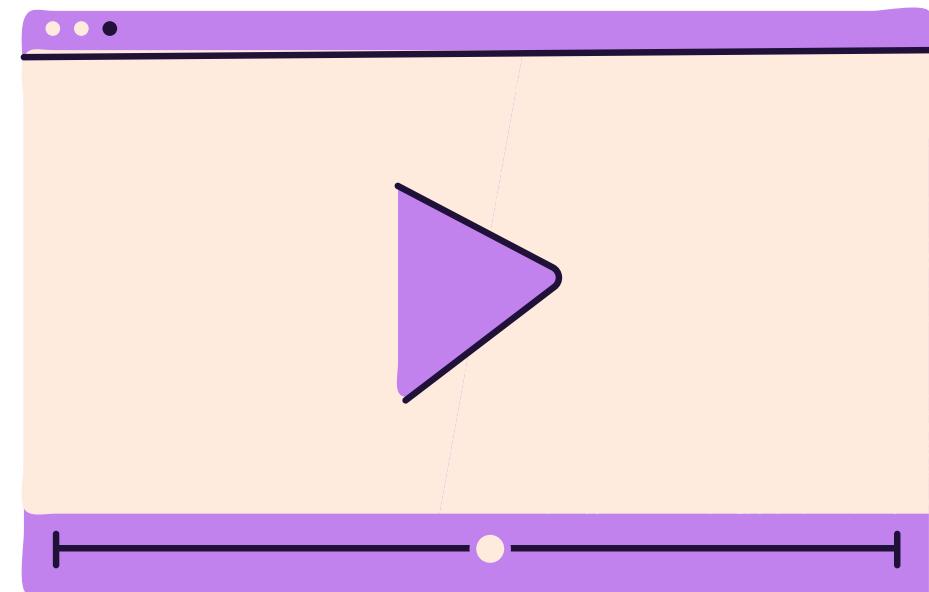
QUERY 3

```
select  
empid,emp_name,DateofBirth  
from employee where gender = 'f'  
and emp_type = 'nur' and  
DateofBirth > '1990-01-01';  
OUTPUT
```

empid	emp_name	DateofBirth
► EM011	Riya	1993-01-10
EM012	Seema	1993-01-01
EM013	Jyoti	1996-01-20
EM014	Poorvi	1991-10-18
NULL	NULL	NULL

QUESTION 4

Write a SQL Query that calculates the cost of the IC PCR test.



QUERY 4

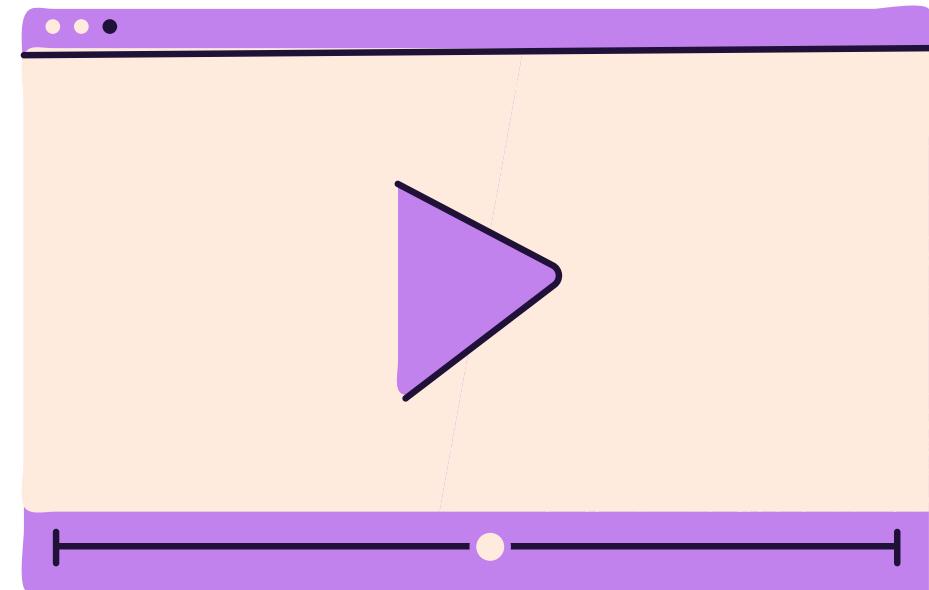
```
select test_cost from test  
where test_name = 'IC PCR Test';
```

OUTPUT

test_cost
1000

QUESTION 5

Write a SQL Query that arranges all the medicines by the cost in ascending order.



QUERY 5

```
select * from medicine  
order by med_cost asc;
```

OUTPUT

	med_id	med_name	med_cost
▶	MED02	Citzin	70
◀	MED07	Paracetamol	100
◀	MED05	Insulin-A	600
◀	MED06	Glynase	700
◀	MED01	Remdesivir	900
◀	MED03	Aspirin	1000
◀	MED04	Wararin	2000
	NULL	NULL	NULL

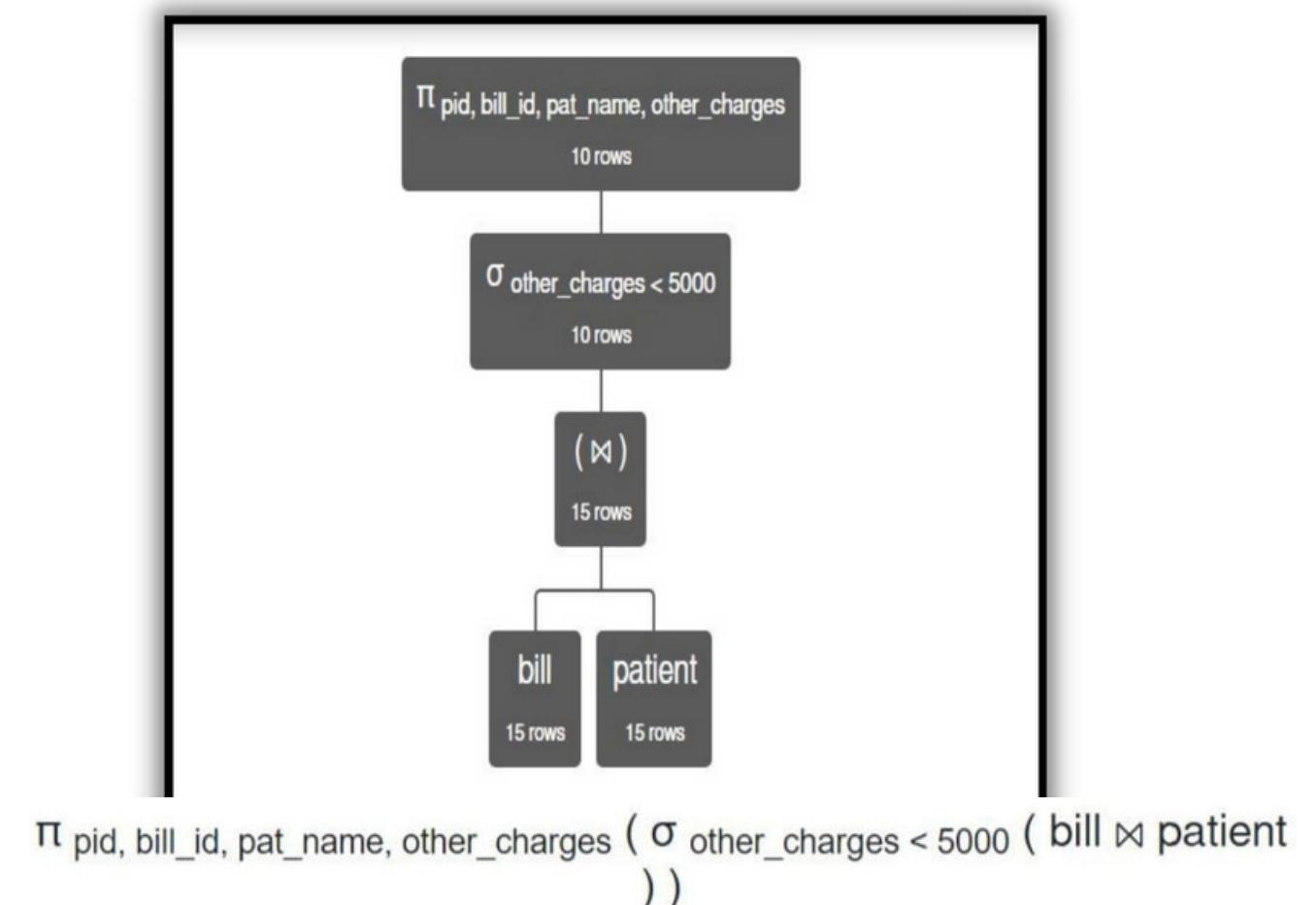


RELATIONAL QUERY

RELATIONAL QUERY 1

1. Write a Query in relational algebra to find all the names and id of all the patients whose other charges are less than 5000.

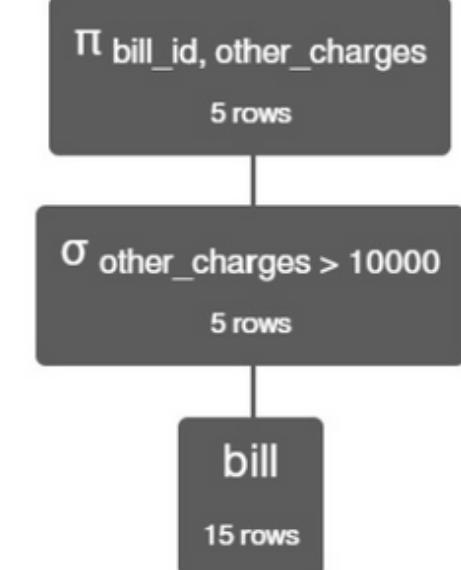
bill.pid	bill.bill_id	patient.pat_name	bill.other_charges
'PAT001'	'BILL01'	'Reena'	1000
'PAT003'	'BILL03'	'Bhuvi'	900
'PAT004'	'BILL04'	'Rakesh'	1200
'PAT006'	'BILL06'	'Laxmi'	2000
'PAT007'	'BILL07'	'Rahul'	1600
'PAT009'	'BILL09'	'Vinita'	1100
'PAT011'	'BILL11'	'Karan'	3000
'PAT012'	'BILL12'	'Aditi'	1200
'PAT013'	'BILL13'	'Harsh'	1000



RELATIONAL QUERY 2

2.) Write a relational algebra Query to fetch out all bills which cost greater than 10000.

bill.bill_id	bill.other_charges
'BILL02'	100000
'BILL05'	55000
'BILL08'	35000
'BILL10'	65000
'BILL15'	90000



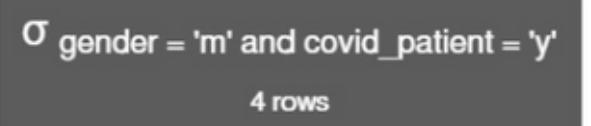
$\Pi_{\text{bill_id, other_charges}} (\sigma_{\text{other_charges} > 10000} (\text{bill}))$

RELATIONAL QUERY 3

3.) Write a relational query that lists all the number of male covid patients.

Π pid, pat_name, Pat_phone_no (σ gender = 'm' and covid_patient = 'y' (patient))

patient.pid	patient.pat_name	patient.Pat_phone_no
'PAT010'	'Tushar'	9123456789
'PAT011'	'Karan'	223456789
'PAT013'	'Harsh'	423456789
'PAT015'	'Deepak'	623456789



Π pid, pat_name, Pat_phone_no (σ gender = 'm' and covid_patient = 'y' (patient))

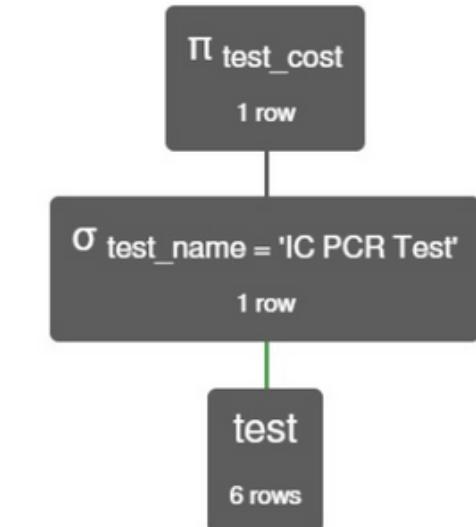
RELATIONAL QUERY 4

4.) Write a SQL Query that calculates the total cost of the IC PCR test.

```
Π test_cost ( σ test_name = 'IC PCR Test' ( test ) )
```

test.test_cost

1000

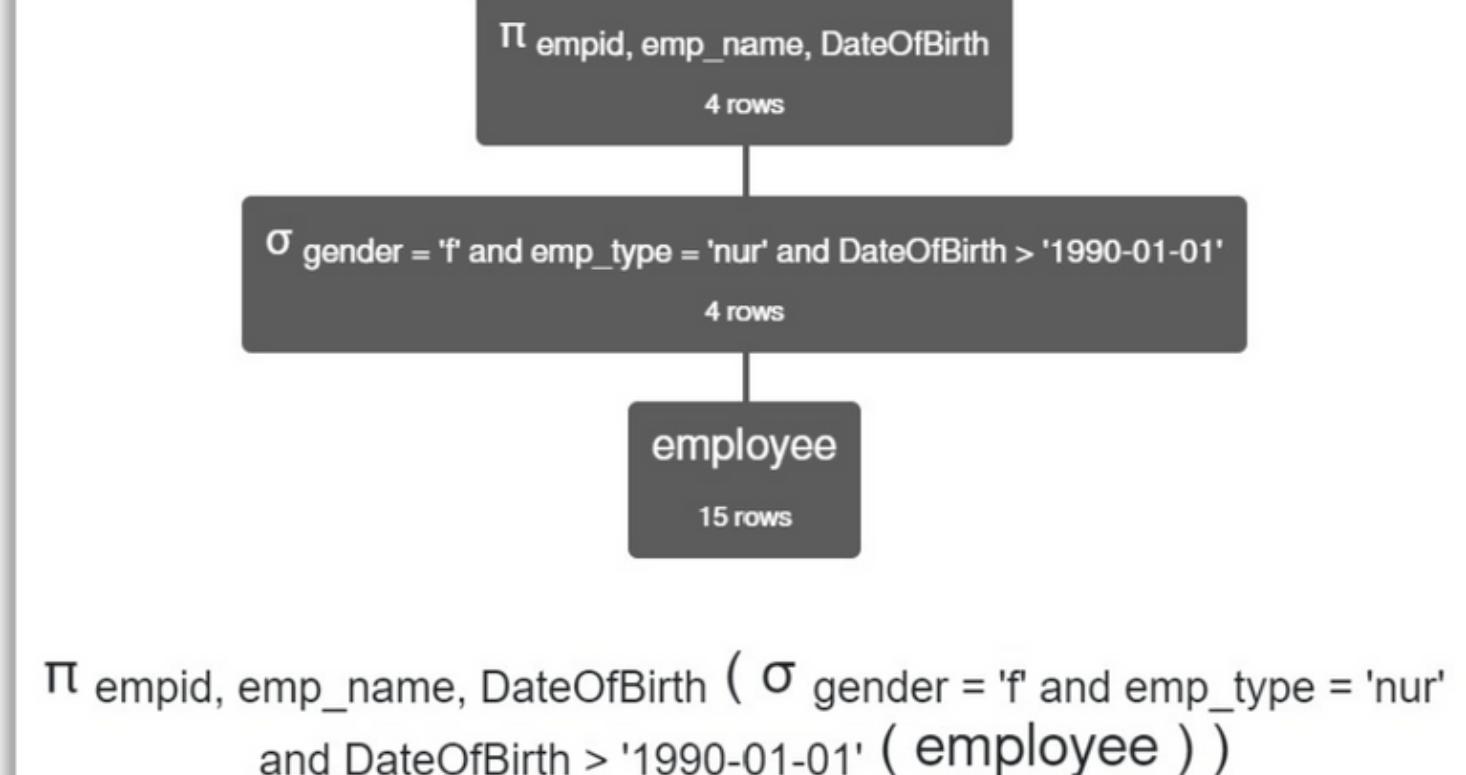


```
Π test_cost ( σ test_name = 'IC PCR Test' ( test ) )
```

RELATIONAL QUERY 5

5.) Write a SQL query that lists employee_id , name , date of birth of all the female nurses born after the year 1990

employee.empid	employee.emp_name	employee.DateOfBirth
'EM011'	'Riya'	'1993-01-10'
'EM012'	'Seema'	'1993-01-01'
'EM013'	'Joyti'	'1996-01-20'
'EM014'	'Poorvi'	'1991-10-18'



- □ ×

THANK YOU

