**STEFFIN JOHN**

**CLASS -MU1**

**HOSPITAL DATABASE**

**EASY**

1. Show first name, last name, and gender of patients who's gender is 'M'

SELECT first\_name,last\_name,gender from patients

where gender = 'M';

1. Show first name and last name of patients who does not have allergies. (null)

select

first\_name,

last\_name

from patients

where allergies is null;

1. Show first name of patients that start with the letter 'C'

select

first\_name

from patients

where first\_name like 'C%';

1. Show first name and last name of patients that weight within the range of 100 to 120 (inclusive)

select

first\_name,

last\_name

from patients

where weight between 100 and 120;

1. Update the patients table for the allergies column. If the patient's allergies is null then replace it with 'NKA'

update patients

set allergies = 'NKA'

where allergies is null;

1. Show first name and last name concatinated into one column to show their full name.

select concat(first\_name,' ',last\_name) as full\_name

from patients;

1. Show first name, last name, and the full province name of each patient.

Example: 'Ontario' instead of 'ON'

select pa.first\_name,pa.last\_name,pr.province\_name from patients pa

join province\_names pr

on pa.province\_id = pr.province\_id;

1. Show how many patients have a birth\_date with 2010 as the birth year.

select count(\*)

from patients

where year(birth\_date) = 2010;

1. Show the first\_name, last\_name, and height of the patient with the greatest height.

select first\_name,last\_name,max(height) as height from patients;

1. Show all columns for patients who have one of the following patient\_ids:  
   1,45,534,879,1000

select \* from patients

where patient\_id in (1,45,534,879,1000);

1. Show the total number of admissions

select count(\*) from admissions;

1. Show all the columns from admissions where the patient was admitted and discharged on the same day.

select \* FROM admissions

WHERE admission\_date = discharge\_date;

1. Show the patient id and the total number of admissions for patient\_id 579.

select patient\_id, COUNT(\*) as total\_admissions

from admissions

where patient\_id = '579';

1. Based on the cities that our patients live in, show unique cities that are in province\_id 'NS'?

select distinct(city) as unique\_cities from patients

where province\_id = 'NS';

1. Write a query to find the first\_name, last name and birth date of patients who has height greater than 160 and weight greater than 70

select

first\_name,

last\_name,

birth\_date

from patients

where height>160 and weight>70;

1. Write a query to find list of patients first\_name, last\_name, and allergies from Hamilton where allergies are not null

select first\_name,last\_name, allergies

from patients

where city = 'Hamilton' and allergies is not null;

1. Based on cities where our patient lives in, write a query to display the list of unique city starting with a vowel (a, e, i, o, u). Show the result order in ascending by city.

SELECT distinct(CITY) from patients

where city like 'a%'

or

city like 'e%'

or

city like 'i%'

or

city like 'o%'

or

city like 'u%'

ORDER BY CITY;

**MEDIUM**

1. Show unique birth years from patients and order them by ascending.

select distinct(Year(birth\_date)) as birth\_year from patients

order by birth\_year;

1. Show unique first names from the patients table which only occurs once in the list.  
     
   For example, if two or more people are named 'John' in the first\_name column then don't include their name in the output list. If only 1 person is named 'Leo' then include them in the output.

select first\_name from patients

group by first\_name

having count(first\_name) = 1;

1. Show patient\_id and first\_name from patients where their first\_name start and ends with 's' and is at least 6 characters long.

select patient\_id,first\_name from patients

where first\_name like 'S\_\_\_\_%S';

1. Show patient\_id, first\_name, last\_name from patients whos diagnosis is 'Dementia'.  
     
   Primary diagnosis is stored in the admissions table.

select pa.patient\_id,pa.first\_name,pa.last\_name from patients pa

join admissions ad

on pa.patient\_id = ad.patient\_id

where ad.diagnosis = 'Dementia';

1. Display every patient's first\_name.  
   Order the list by the length of each name and then by alphabetically

select first\_name from patients

order by len(first\_name), first\_name asc;

1. Show the total amount of male patients and the total amount of female patients in the patients table.  
   Display the two results in the same row.

select

(select count(\*) from patients where gender = 'M') as male\_count,

(select count(\*) from patients where gender = 'F') as female\_count;

1. Show first and last name, allergies from patients which have allergies to either 'Penicillin' or 'Morphine'. Show results ordered ascending by allergies then by first\_name then by last\_name.

select first\_name,last\_name,allergies from patients

where allergies = 'Penicillin' or allergies = 'Morphine'

order by allergies,first\_name,last\_name;

1. Show patient\_id, diagnosis from admissions. Find patients admitted multiple times for the same diagnosis

select patient\_id,diagnosis

from admissions

group by diagnosis,patient\_id

having count(diagnosis)!=1;

1. Show the city and the total number of patients in the city.  
   Order from most to least patients and then by city name ascending.

select city,count(\*) from patients

group by city

order by count(\*) desc,city asc;

1. Show first name, last name and role of every person that is either patient or doctor.  
   The roles are either "Patient" or "Doctor"

SELECT first\_name,last\_name,'Patient' AS role FROM patients

UNION ALL

SELECT first\_name,last\_name,'Doctor' FROM doctors;

1. Show all allergies ordered by popularity. Remove NULL values from query.

select allergies,count(allergies)as tot from patients

where allergies is not null

group by allergies

order by tot desc;

1. Show all patient's first\_name, last\_name, and birth\_date who were born in the 1970s decade. Sort the list starting from the earliest birth\_date

select first\_name,last\_name,birth\_date from patients

where year(birth\_date) between 1970 and 1979

order by birth\_date asc;

1. We want to display each patient's full name in a single column. Their last\_name in all upper letters must appear first, then first\_name in all lower case letters. Separate the last\_name and first\_name with a comma. Order the list by the first\_name in decending order  
   EX: SMITH,jane

select concat(upper(last\_name),',',lower(first\_name)) as full\_name from patients

order by first\_name desc;

1. Show the province\_id(s), sum of height; where the total sum of its patient's height is greater than or equal to 7,000.

select province\_id,sum(height)as sum\_height from patients

group by province\_id

having sum(height)>=7000;

1. Show all of the days of the month (1-31) and how many admission\_dates occurred on that day. Sort by the day with most admissions to least admissions.

select day(admission\_date) day\_number,count(patient\_id) number\_of\_admissions

from admissions

group by day\_number

order by number\_of\_admissions desc;

1. Show all columns for patient\_id 542's most recent admission\_date.

select \* from admissions

group by patient\_id

having

patient\_id = 542 and max(admission\_date);

1. Show patient\_id, attending\_doctor\_id, and diagnosis for admissions that match one of the two criteria:  
   1. patient\_id is an odd number and attending\_doctor\_id is either 1, 5, or 19.  
   2. attending\_doctor\_id contains a 2 and the length of patient\_id is 3 characters.

select patient\_id,attending\_doctor\_id,diagnosis from admissions

where patient\_id %2 !=0 and attending\_doctor\_id in(1,5,19)

or

attending\_doctor\_id like '%2%' and len(patient\_id)=3;

1. Show first\_name, last\_name, and the total number of admissions attended for each doctor.  
     
   Every admission has been attended by a doctor.

Show first\_name, last\_name, and the total number of admissions attended for each doctor.  
  
Every admission has been attended by a doctor.

1. For each doctor, display their id, full name, and the first and last admission date they attended.

SELECT doctor\_id,

CONCAT(first\_name,' ',last\_name) AS full\_name,

MIN(admission\_date) AS first\_addmission,

MAX(admission\_date) as last\_admissions

FROM admissions a

JOIN doctors d

ON a.attending\_doctor\_id = d.doctor\_id

GROUP BY doctor\_id;

select

doctor\_id,

first\_name || ' ' || last\_name as full\_name,

min(admission\_date) as first\_admission\_date,

max(admission\_date) as last\_admission\_date

from admissions a

join doctors ph on a.attending\_doctor\_id = ph.doctor\_id

group by doctor\_id;

1. Display the total amount of patients for each province. Order by descending.

select pn.province\_name, count(p.patient\_id) as patient\_counts from patients as p

join province\_names as pn

on p.province\_id=pn.province\_id

group by pn.province\_name

order by patient\_counts desc;

1. For every admission, display the patient's full name, their admission diagnosis, and their doctor's full name who diagnosed their problem.

select concat(p.first\_name,' ',p.last\_name) as full\_name,a.diagnosis,

concat(d.first\_name,' ',d.last\_name) as doctor\_name

from patients p

join admissions a

on p.patient\_id = a.patient\_id

join doctors d

on a.attending\_doctor\_id = d.doctor\_id;

1. display the number of duplicate patients based on their first\_name and last\_name.

select first\_name,last\_name,count(patient\_id) no\_of\_duplicates from patients

group by first\_name,last\_name

having

count(patient\_id)>1;

1. Display patient's full name,  
   height in the units feet rounded to 1 decimal,  
   weight in the unit pounds rounded to 0 decimals,  
   birth\_date,  
   gender non abbreviated.  
     
   Convert CM to feet by dividing by 30.48.  
   Convert KG to pounds by multiplying by 2.205.

SELECT CONCAT(first\_name,' ',last\_name) AS full\_name,

ROUND((height/30.48),1) AS height\_cm, ROUND((weight\*2.205)) AS weight\_pounds,

birth\_date,

CASE

WHEN gender = 'M' THEN 'MALE'

when gender = 'F' then 'FEMALE'

end as gender

from patients;

select

concat(first\_name, ' ', last\_name) AS 'patient\_name',

ROUND(height / 30.48, 1) as 'height "Feet"',

ROUND(weight \* 2.205, 0) AS 'weight "Pounds"', birth\_date,

CASE

WHEN gender = 'M' THEN 'MALE'

ELSE 'FEMALE'

END AS 'gender\_type'

from patients

**HARD**

1. Show all of the patients grouped into weight groups.  
   Show the total amount of patients in each weight group.  
   Order the list by the weight group decending.  
     
   For example, if they weight 100 to 109 they are placed in the 100 weight group, 110-119 = 110 weight group, etc.

select count(patient\_id) as patients\_in\_group,

floor(weight/10) \*10 as weight\_group from patients

group by weight\_group

order by weight\_group desc;

1. Show patient\_id, weight, height, isObese from the patients table.  
     
   Display isObese as a boolean 0 or 1.  
     
   Obese is defined as weight(kg)/(height(m)2) >= 30.  
     
   weight is in units kg.  
     
   height is in units cm.

select patient\_id,weight,height,

case

when weight/(power(height/100.0,2)) >= 30 then 1

else 0

end as isobese

from patients;

select patient\_id,weight,height,

case

when weight/(power(height\*0.01,2)) >= 30 then 1 --- why not working

else 0

end as isobese

from patients;

1. Show patient\_id, first\_name, last\_name, and attending doctor's specialty.  
   Show only the patients who has a diagnosis as 'Epilepsy' and the doctor's first name is 'Lisa'  
     
   Check patients, admissions, and doctors tables for required information.

select p.patient\_id,p.first\_name,p.last\_name,d.specialty from patients p

join admissions a

on p.patient\_id = a.patient\_id

join doctors d

on a.attending\_doctor\_id = d.doctor\_id

where a.diagnosis = 'Epilepsy' and d.first\_name = 'Lisa';

1. All patients who have gone through admissions, can see their medical documents on our site. Those patients are given a temporary password after their first admission. Show the patient\_id and temp\_password.  
     
   The password must be the following, in order:  
   1. patient\_id  
   2. the numerical length of patient's last\_name  
   3. year of patient's birth\_date

select distinct(p.patient\_id), concat(p.patient\_id,len(p.last\_name),year(p.birth\_date))

as temp\_password

from patients p

join admissions a

on p.patient\_id = a.patient\_id;

1. Each admission costs $50 for patients without insurance, and $10 for patients with insurance. All patients with an even patient\_id have insurance.  
     
   Give each patient a 'Yes' if they have insurance, and a 'No' if they don't have insurance. Add up the admission\_total cost for each has\_insurance group.

select

case

when patient\_id %2 = 0 then 'yes'

else 'No'

end as has\_insurance,

sum(case

when patient\_id %2 = 0 then '10'

else '50'

end) as cost\_after\_insurance

from admissions

group by has\_insurance

1. Show the provinces that has more patients identified as 'M' than 'F'. Must only show full province\_name

select pn.province\_name

from patients p

join province\_names pn

on p.province\_id = pn.province\_id

group by pn.province\_name

having count(case when p.gender = 'M' then 1 end) > count(case when p.gender = 'F' then 1 end);

1. We are looking for a specific patient. Pull all columns for the patient who matches the following criteria:  
   - First\_name contains an 'r' after the first two letters.  
   - Identifies their gender as 'F'  
   - Born in February, May, or December  
   - Their weight would be between 60kg and 80kg  
   - Their patient\_id is an odd number  
   - They are from the city 'Kingston'

Select \* from patients

Where

First\_name like ‘\_\_r%’

And gender = ‘F’

And Month(birth\_date) in (2,512)

And weight between 60 and 80

And patient\_id %2 = 1

And city = ’Kingston’;

1. Show the percent of patients that have 'M' as their gender. Round the answer to the nearest hundreth number and in percent form.

Select

Concat (round (sum (

Case when gender = 'M' then 1

Else 0

End) \* 100.00/ count (\*),2),'%')

From patients;

SELECT CONCAT(

ROUND(

(

SELECT COUNT(\*)

FROM patients

WHERE gender = 'M'

) / CAST(COUNT(\*) as float),

4

) \* 100,

'%'

) as percent\_of\_male\_patients

FROM patients;

SELECT

round(100 \* avg(gender = 'M'), 2) || '%' AS percent\_of\_male\_patients

FROM

patients;

SELECT

CONCAT(ROUND(SUM(gender='M') / CAST(COUNT(\*) AS float), 4) \* 100, '%')

FROM patients;

1. For each day display the total amount of admissions on that day. Display the amount changed from the previous date.

select admission\_date,count(admission\_date)admission\_day,

count(admission\_date)-lag(count(admission\_date)) over(order by admission\_date)as count\_date

from admissions

group by admission\_date

select admission\_date,count(admission\_date)admission\_day,

count(admission\_date)-lag(count(admission\_date)) over(order by admission\_date)

from admissions

group by admission\_date

WITH admission\_counts\_table AS (

SELECT admission\_date, COUNT(patient\_id) AS admission\_count

FROM admissions

GROUP BY admission\_date

ORDER BY admission\_date DESC

)

select

admission\_date,

admission\_count,

admission\_count - LAG(admission\_count) OVER(ORDER BY admission\_date) AS admission\_count\_change

from admission\_counts\_table

1. Sort the province names in ascending order in such a way that the province 'Ontario' is always on top.

select province\_name from province\_names

order by

(case

when province\_name = 'Ontario' then 0 else 1

end),

province\_name;

**NORTHWIND DATABASE**

**EASY**

1. Show the category\_name and description from the categories table sorted by category\_name.

select category\_name,description from categories

group by category\_name;

1. Show all the contact\_name, address, city of all customers which are not from 'Germany', 'Mexico', 'Spain'

select contact\_name,address,city from customers

where country not in('Germany', 'Mexico', 'Spain');

1. Show order\_date, shipped\_date, customer\_id, Freight of all orders placed on 2018 Feb 26

select order\_date,shipped\_date,customer\_id,freight from orders

where order\_date = '2018-02-26';

1. Show the employee\_id, order\_id, customer\_id, required\_date, shipped\_date from all orders shipped later than the required date

select employee\_id,order\_id,customer\_id,required\_date,shipped\_date from orders

where shipped\_date > required\_date;

1. Show all the even numbered Order\_id from the orders table

select order\_id from orders

where mod(order\_id,2) = 0;

1. Show the city, company\_name, contact\_name of all customers from cities which contains the letter 'L' in the city name, sorted by contact\_name

select city,company\_name,contact\_name from customers

where city like '%L%'

group by contact\_name;

1. Show the company\_name, contact\_name, fax number of all customers that has a fax number. (not null)

select company\_name,contact\_name,fax from customers

where fax is not null;

1. Show the first\_name, last\_name of the most recently hired employee.

select first\_name,last\_name, max(hire\_date) as hire\_date from employees;

1. Show the average unit price rounded to 2 decimal places, the total units in stock, total discontinued products from the products table.

select round(avg(unit\_price),2) as avg\_unit\_price,sum(units\_in\_stock),sum(discontinued)

from products;

**MEDIUM**

1. Show the ProductName, CompanyName, CategoryName from the products, suppliers, and categories table

select product\_name,company\_name,category\_name from categories c

join products p

on c.category\_id = p.category\_id

join suppliers s

on p.supplier\_id = s.supplier\_id;

1. Show the category\_name and the average product unit price for each category rounded to 2 decimal places

select category\_name,round(avg(unit\_price),2) as avg\_unit\_price from categories c

join products p

on c.category\_id = p.category\_id

group by category\_name;

1. Show the city, company\_name, contact\_name from the customers and suppliers table merged together.  
   Create a column which contains 'customers' or 'suppliers' depending on the table it came from.

select city,company\_name,contact\_name,'customers' as relationship from customers

union

select city,company\_name,contact\_name,'suppliers' from suppliers;

select city,company\_name,contact\_name,'customers' as relationship from customers

union all

select city,company\_name,contact\_name,'suppliers' as relationship from suppliers

**HARD**

1. Show the employee's first\_name and last\_name, a "num\_orders" column with a count of the orders taken, and a column called "Shipped" that displays "On Time" if the order shipped on time and "Late" if the order shipped late.  
     
   Order by employee last\_name, then by first\_name, and then descending by number of orders.

select e.first\_name,e.last\_name, count(o.order\_id) as num\_orders,

case

when o.shipped\_date<o.required\_date then 'On Time'

else 'Late'

end as Shipped

from employees e

join orders o

on e.employee\_id = o.employee\_id

group by e.first\_name,e.last\_name,Shipped

order by e.last\_name,e.first\_name,num\_orders desc;