FINAL SQL PROJECT HOSPITAL DATABASE

Sec - 2

Project Done by:
Vishnu Priya Ashok Kumar – 8767381

Course Info: INFO8075 SQL and Data Analysis
Professor: Yichen Zhang
Date: 14/12/2022
Conestoga College
Doon Campus

Table of Contents

1 INTRODUCTION

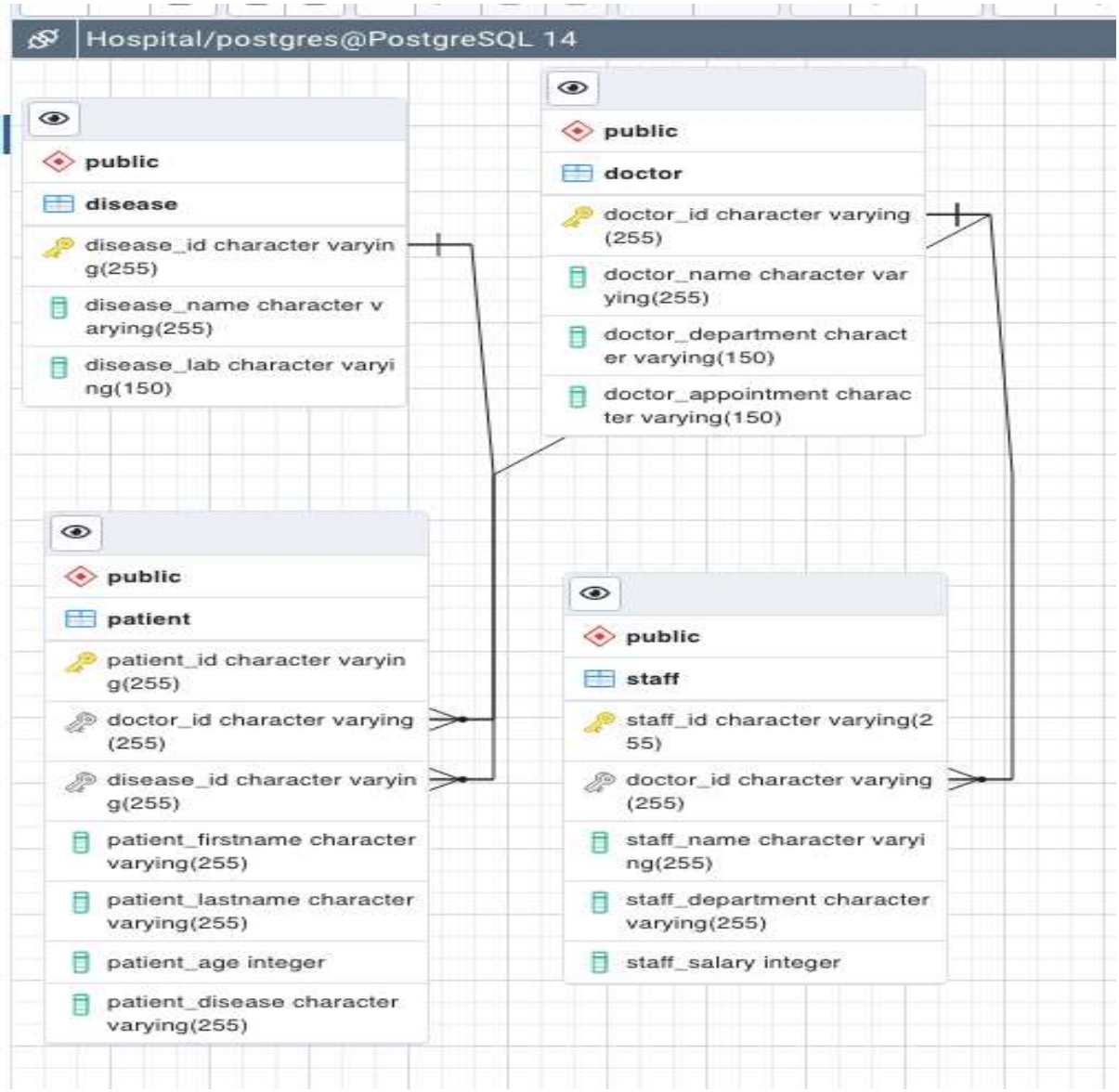
In this project we have decided to use SQL database queries to work on hospital administration. In our project we used an existing developed database to gain more insights into the dataset by executing syntax and queries which we recently acquired during the course of the program. This database is used in the hospital to treat a variety of diseases departments and accommodates a large number of patients, workers, and physicians.

The four tables in this hospital database are needed to manage the hospital:

- Staffs: This table provides information on the hospital staffs, including their names, departmental assignments, staff identification numbers, and salaries. It also shows which staffs report to which doctors.
- Doctor: Information on the doctors employed by this hospital, including name, ID, special department, and appointment times, is shown in this table.
- Patient: Information about each patient, including name, identification number, age, disease, doctor's ID, and location, is explained in this table.
- Diseases: This table includes details on the name, ID, and department of each illness.

The values for all the tables are copied from the csv files to each of the tables.

2 ER DIAGRAM FOR HOSPITAL DATABASE



3 IMPLEMENTING VARIOUS SQL QUERY FUNCTIONS

3.1 Query to Display Patient's Rank Based on Their Assigned Doctor

***Syntax used*:**

```
SELECT CONCAT(patient_firstname, ' ', patient_lastname) AS patient_NAME,
d.doctor_name, d.doctor_department,patient_age,
RANK() OVER (
        PARTITION BY p.doctor_id ORDER BY patient_age) AS patient_Rank
FROM patient p
INNER JOIN doctor d
ON p.doctor_id = d.doctor_id;
```

```
50
51  -- 1. QUERY TO DISPLAY PATIENT'S RANK BASED ON THEIR ASSIGNED DOCTOR
52  SELECT CONCAT(patient_firstname, ' ', patient_lastname) AS patient_NAME,
53  d.doctor_name, d.doctor_department,patient_age,
54  RANK() OVER (
55        PARTITION BY p.doctor_id ORDER BY patient_age) AS patient_Rank
56  FROM patient p
57  inner join doctor d
58  on p.doctor_id = d.doctor_id;
59
```

Data output | Messages | Notifications

	patient_name text	doctor_name character varying (255)	doctor_department character varying (150)	patient_age integer	patient_rank bigint
1	Allen Clay	Adam	Anesthesiology	28	1
2	Amanda Depp	Adam	Anesthesiology	40	2
3	Rachel John	Thomas	General surgeon	21	1
4	Ken Liam	Landon	Pediatrics	45	1
5	Joel John	Candace	Obstetrician/gynecolo..	88	1
6	Safwan Farrell	Carole	Gastroenterology	60	1
7	Sophie Santa..	Geraldo	Respirology	50	1
8	Carol Glen	Ben	Ophthalmology	33	1
9	Rose Jack	Ben	Ophthalmology	40	2

Total rows: 20 of 20 Query complete 00:00:00.060

3.2 Query to Display Sum of Patients By Dividing Their Age Into Bucket of 5 Ranging Between 15 to 90

Syntax used:

```
SELECT BUCKET, SUM(patient_count) OVER (ORDER BY BUCKET) FROM
(
SELECT WIDTH_BUCKET(patient_age, 15, 90, 5) AS BUCKET,
COUNT(*) AS patient_count
FROM patient
GROUP BY BUCKET
ORDER BY BUCKET ) S;
```

```
60
61  -- 2. QUERY TO DISPLAY SUM OF PATIENTS BY DIVIDING THEIR AGE INTO BUCKET OF 5 RANGING BETWEEN 15 TO 90
62  SELECT BUCKET, SUM(patient_count) OVER (ORDER BY BUCKET) FROM
63  (
64  SELECT WIDTH_BUCKET(patient_age, 15, 90, 5) AS BUCKET,
65  COUNT(*) AS patient_count
66  FROM patient
67  GROUP BY BUCKET
68  ORDER BY BUCKET ) S;
69
```

Data output | Messages | Notifications

	bucket integer	sum numeric
1	1	5
2	2	10
3	3	14
4	4	18
5	5	20

3.3 Query to display patient name and age and categorizing patient into elderly, youth, and teenager

***Syntax used*:**

```
SELECT Patient_id,Patient_FirstName,Patient_lastName,Patient_Age,
CASE WHEN patient_Age >=50 THEN 'Elderly'
          WHEN patient_Age between 23 and 48 THEN 'youth'
           ELSE'Teenager'
           END AS PatientAge_range
FROM Patient
```

```
88
89  Select Patient_id,Patient_FirstName,Patient_lastName,Patient_Age,
90  Case when patient_Age >=50 then 'Elderly'
91            When patient_Age between 23 and 48 then 'youth'
92            Else 'Teenager'
93            end as PatientAge_range
94  From Patient
95
96
```

Data output | Messages | Notifications

patient_id [PK] character varying (255)	patient_firstname character varying (255)	patient_lastname character varying (255)	patient_age integer	patientage_range text
501	Amanda	Depp	40	youth
502	Carol	Glen	33	youth
503	Nancy	jose	23	youth
504	Alan	Fin	65	Elderly
505	Dill	Ben	48	youth
506	Adam	Smith	75	Elderly
507	Tessy	Jacob	62	Elderly
508	Rachel	John	21	Teenager
509	Ken	Liam	45	youth

3.4 Display Disease Count Most Common in Lab

***Syntax used*:**

```
WITH top_diseaselab as (
    SELECT Disease.disease_lab, count(patient.disease_id) as disease_count
FROM Disease, Patient
WHERE Disease.disease_id =patient.disease_id
GROUP BY Disease.disease_lab
)
SELECT Disease.*, top_diseaselab.Disease_count
FROM Disease
JOIN top_diseaselab
ON Disease.disease_lab =top_diseaselab.disease_lab
ORDER BY Disease_count Desc;
```

```
97
98  With top_diseaselab as (
99          Select Disease.disease_lab, count(patient.disease_id) as disease_count
100 From Disease, Patient
101 Where Disease.disease_id =patient.disease_id
102 Group by Disease.disease_lab
103 )
104 Select Disease.*, top_diseaselab.Disease_count
105 From Disease
106 Join top_diseaselab
107 On Disease.disease_lab =top_diseaselab.disease_lab
108 Order by Disease_count Desc;
109
```

Data output | Messages | Notifications

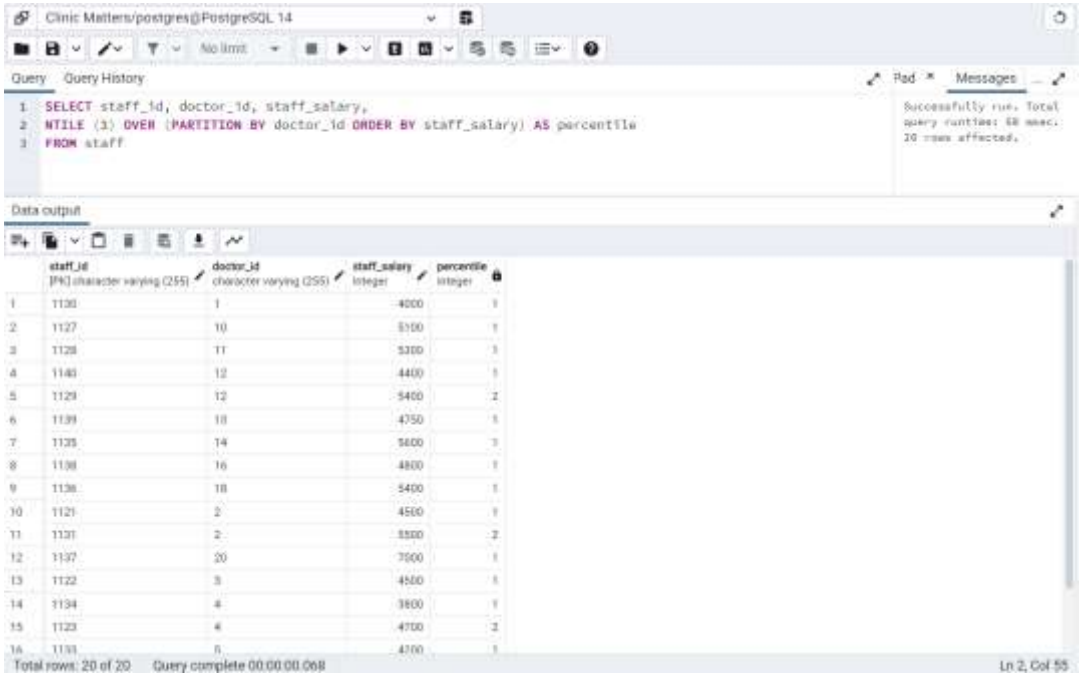
	disease_id [PK] character varying (255)	disease_name character varying (255)	disease_lab character varying (150)	disease_count bigint
1	1016	Legionella	Blood Test	9
2	1003	Hepatitis	Blood Test	9
3	1004	Lyme Disease	Blood Test	9
4	1005	Meningococcal Disease	Blood Test	9
5	1002	Asthma	Blood Test	9
6	1013	Influenza	Blood Test	9
7	1014	Varicella	Blood Test	9
8	1007	Norwalk Virus	Infection screening	4
9	1017	Rubella	Infection screening	4

Total rows: 20 of 20 Query complete 00:00:00.177

3.5 Query to create a percentile on salary based on doctor's ID

***Syntax used*:**

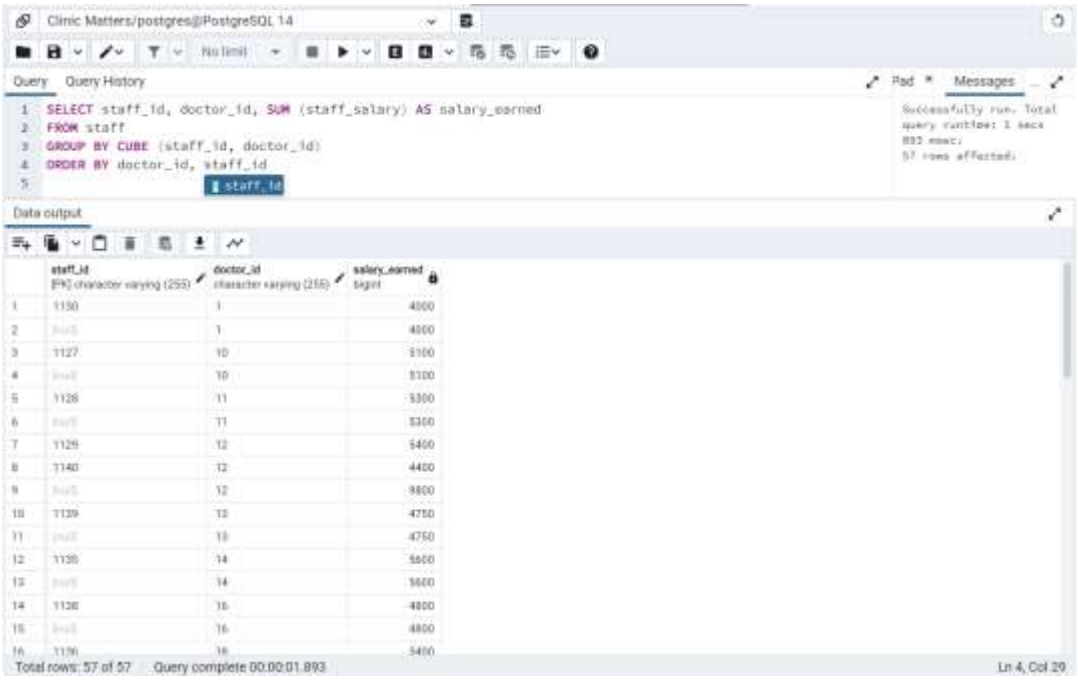
SELECT staff_id, doctor_id, staff_salary,
NTILE (3) OVER (PARTITION BY doctor_id ORDER BY staff_salary) AS percentile
FROM staff;



3.6 Query to Identify the Subtotal and Total of Salaries Paid to Doctors According to their ID

Syntax used:

```
SELECT staff_id, doctor_id, SUM (staff_salary) AS salary_earned
FROM staff
GROUP BY CUBE (staff_id, doctor_id)
ORDER BY doctor_id, staff_id;
```



3.7 Query to Display Doctor's and Staff Tables Using Join and Subqueries Statements

Syntax used:

```
SELECT d.doctor_id, d.doctor_name
FROM doctor d
INNER JOIN staff s
ON d.doctor_id = s.doctor_id
WHERE d.doctor_id=ANY(SELECT doctor_id FROM staff)
AND s.staff_salary >=2000;
```

```
SELECT d.doctor_id, d.doctor_name
FROM doctor d
inner join staff s
on d.doctor_id = s.doctor_id
where d.doctor_id=ANY(SELECT doctor_id FROM staff)
AND s.staff_salary >=2000;
```

ata Output | Messages | Notifications

doctor_id [PK] character varying (255)	doctor_name character varying (255)
1	Adam
2	Ben
2	Ben
3	Charlie
4	Diana
4	Diana
5	Edwin
5	Edwin
6	Dave

otal rows: 20 of 20 Query complete 00:00:00.223

3.8 Query to Display Aggregation and Joins of Staff and Patient Table

Syntax used:

SELECT A.doctor_id, B.staff_name,
COUNT (DISTINCT B.staff_id) AS staff_count,
SUM(B.staff_salary) as Total_salary
FROM patient A
INNER JOIN staff B
ON A.doctor_id=B.doctor_id
GROUP BY A.doctor_id, B.staff_name

Query Query History

```
1  SELECT A.doctor_id, B.staff_name,
2  COUNT (DISTINCT B.staff_id ) as staff_count,
3  sum(B.staff_salary) as Total_salary
4  from patient A
5  inner join staff B
6  ON A.doctor_id=B.doctor_id
7  GROUP BY A.doctor_id, B.staff_name
8
```

Data Output Messages Notifications

	doctor_id character varying (255)	staff_name character varying (255)	staff_count bigint	total_salary bigint
1	1	Alicia	1	8000
2	10	Adam	1	5100
3	11	Scott	1	5300
4	12	Boyer	1	4400
5	12	Natalie	1	5400
6	13	Weaver	1	4750
7	14	Ferdinand	1	5600
8	2	Brek	1	9000
9	2	Edgar	1	11000
10	3	Kathy	1	18000
11	4	Bruce	1	4700
12	4	Cleo	1	3800
13	5	Cory	1	4200
14	5	Liza	1	4800
15	6	Jenifer	1	9600
16	7	Allen	1	5200
17	9	Maria	1	5200

3.9 Query to Display Doctor's name, Department Names, Staffs and Departments Using the 'Left Join'

***Syntax used*:**

SELECT doctor_name, doctor_department, staff_name, staff_department
FROM doctor
LEFT JOIN staff
ON doctor.doctor_id = staff.doctor_id;

```
59
60  SELECT doctor_name, doctor_department, staff_name, staff_department
61  FROM doctor
62  LEFT JOIN staff
63  ON doctor.doctor_id = staff.doctor_id;
64
65
```

Data output | Messages | Notifications

	doctor_name character varying (255)	doctor_department character varying (150)	staff_name character varying (255)	staff_department character varying (255)
1	Ben	Ophthalmology	Brek	Ophthalmology
2	Charlie	Pediatrics	Kathy	Pediatrics
3	Diana	Otolaryngology	Bruce	Otolaryngology
4	Edwin	Clinical Immunology	Liza	Clinical Immunology
5	Dave	Clinical Immunology	Jenifer	Clinical Immunology
6	Paul	Obstetrician/gynecolo...	Maria	Cardiologists
7	Thomas	General surgeon	Adam	General surgeon
8	Landon	Pediatrics	Scott	General surgeon
9	Candace	Obstetrician/gynecolo...	Natalie	Obstetrician/gynecolo...
10	Adam	Anesthesiology	Alicia	Anesthetist
11	Ben	Ophthalmology	Edgar	Ophthalmology
12	Jenifer	General surgeon	Allen	Circulating Nurse
13	Edwin	Clinical Immunology	Cory	Clinical Immunology
14	Diana	Otolaryngology	Cleo	Otolaryngology

3.10 Query using ‘Replace’ to Change Doctor Appointment TimE

Syntax used:

SELECT doctor_appointment, REPLACE(doctor_appointment, '10AM - 1PM', '10AM - 12PM')
FROM doctor;

```
75
76  SELECT doctor_appointment, REPLACE(doctor_appointment, '10AM - 1PM', '10AM - 12PM')
77  FROM doctor;
78
```

Data output | Messages | Notifications

	doctor_appointment character varying (150)	replace text
1	10AM - 1PM	10AM - 12PM
2	5PM - 8PM	5PM - 8PM
3	2PM - 5PM	2PM - 5PM
4	7PM - 10PM	7PM - 10PM
5	8AM - 11AM	8AM - 11AM
6	5PM - 7PM	5PM - 7PM
7	12PM - 3PM	12PM - 3PM
8	10 AM - 2PM	10 AM - 2PM
9	1PM - 4PM	1PM - 4PM

3.11 Query to Change Patient Disease Datatype Using The 'Cast' syntax

***Syntax used*:**

SELECT patient_firstname, CAST (patient_disease AS CHAR(30))
Char_patient_disease FROM patient;

```
68
69  SELECT patient_firstname, CAST (patient_disease AS CHAR(30))
70  Char_patient_disease FROM patient;
71
72
```

Data output | Messages | Notifications

	patient_firstname character varying (255)	char_patient_disease character (30)
1	Amanda	Heart Disease
2	Carol	Asthma
3	Nancy	Skin Disease
4	Alan	Stomach Disease
5	Dill	Eye Disease
6	Adam	Head Disease
7	Tessy	Neck Pain
8	Rachel	Bone Pain
9	Ken	Knee joint Pain
10	Joel	Spine Disease
11	Rose	Lung infection
12	Dave	Leptospirosis
13	Annika	Brucellosis
14	Maisey	Skin Disease
15	Aiden	Head Disease

3.12 Query To Display Patient based on Skin Disease

***Syntax used*:**

SELECT patient_id, patient_firstname, patient_lastname
FROM patient
WHERE patient_disease =’Skin Disease’

```
137
138  SELECT
139  patient_id,patient_firtsname, patient_lastname
140
141  FROM patient
142
143  WHERE patient_disease  = 'Skin Disease'
144
145
```

Data Output | Explain | Messages

	patient_id [PK] character varying (255)	patient_firtsname character varying (255)	patient_lastname character varying (255)
1	503	Nancy	jose
2	514	Maisey	Schmidt
3	518	Jana	Hoffman
4	519	Honor	Oneill

3.13 Query to Select from the Doctor Department Specialised in Clinical Immunology

***Syntax used*:**

SELECT doctor_id, doctor_name, doctor _department

FROM doctor

WHERE doctor_department = 'Clinical Immunology'

```
137  select *from patient
138
139
140  SELECT
141  doctor_id, doctor_name, doctor _department
142
143  FROM doctor
144
145  WHERE doctor_department  = 'Clinical Immunology'
146
```

Data Output | Explain | Messages

	doctor_id [PK] character varying (255)	doctor_name character varying (255)	_department doctor
1	5	Edwin	(5,Edwin,"Clinical Immunology","8AM - 11AM")
2	6	Dave	(6,Dave,"Clinical Immunology","5PM - 7PM")

3.14 Using the Case When to Identify Different Disease Names and How They Should Be Handled

***Syntax used*:**

```
SELECT disease_lab, disease_name,
Case when disease_lab ='Infection Screening' then 'Quarantine/CDC'
            Else 'No Drill'
 end as diseaselab_range
FROM disease
```

```
141  SELECT disease_lab, disease_name,
142  Case when disease_lab ='Infection Screening' then 'Quarantine/CDC'
143              Else 'No Drill'
144   end as diseaselab_range
145  FROM disease
146
147
```

Data Output | Explain | Messages

	disease_lab character varying (150)	disease_name character varying (255)	diseaselab_range text
1	ECG Monitoring	Heart Disease	No Drill
2	Blood Test	Asthma	No Drill
3	Blood Test	Hepatitis	No Drill
4	Blood Test	Lyme Disease	No Drill
5	Infection Screening	Plague	Quarantine/CDC
6	Infection Screening	Norwalk Virus	Quarantine/CDC
7	Infection Screening	Pneumococcal Disease	Quarantine/CDC

12	Blood Test	Influenza	No Drill
13	Blood Test	Varicella	No Drill
14	Infection Scrrening	Dengue Fever	No Drill
15	Blood Test	Legionella	No Drill
16	Infection Screening	Rubella	Quarantine/CDC
17	Blood and Infection Test	Tularemia	No Drill
18	Infection Screening	Escherichia coli	Quarantine/CDC

3.15 Using the Max Function with having Clause and Select the Highest Payment

***Syntax used*:**

```
SELECT staffs_id, staffs_name,
MAX (staffs_salary)
FROM staffs
GROUP BY staffs_id, staffs_name
HAVING MAX(staffs_salary) >'5000'
```

```
140  SELECT staffs_id, staffs_name,
141  MAX (staffs_salary)
142  FROM staffs
143  GROUP BY staffs_id, staffs_name
144  HAVING MAX(staffs_salary) >'5000'
145
```

Data Output | Explain | Messages

	staffs_id [PK] character varying (255)	staffs_name character varying (255)	max text
1	1137	Dennis	7000
2	1131	Edgar	5500
3	1127	Adam	5100
4	1132	Allen	5200
5	1128	Scott	5300
6	1129	Natalie	5400
7	1135	Ferdinand	5600
8	1126	Mariah	5200
9	1136	Davidson	5400

4 Goal of the project

We may analyse the following using the hospital database set:

I. To display the ranking of the patient based on their assigned doctor

II. To display the total number of patients by separating their age into a bucket of 5 ranging from 15 to 90

III. To display the patient's name and age, as well as to categorize the patient as older, younger, or teenager
IV. To find the most regular disease count in the lab
V. To create a percentile of salary based on a doctor's id query
VI. To identify the subtotal and total of salary payments made to doctors in accordance with their id
VII. To display doctor's and staff tables with join and subqueries statements
VIII. To show joins and aggregations of the staff and patient table
IX. To display the name of the doctor, the department names, the staffs, and the departments.
X. To change doctor appointment times with a query using "replace"
XI. To change patient disease datatype
XII. To display patient skin disease
XIII. To select the doctors within the clinical immunology department
XIV. To identify various disease names and how to manage them
XV. To find the highest payment for the staffs