UNIVERSITI TUNKU ABDUL RAHMAN

Faculty of Information and Communication Technology (FICT)

UCCD2203 DATABASE SYSTEMS
Group Assignment
(individual submission)
Session 202101

Deadline: Saturday 27 March 2021 (Week 10)
Time: before 5.00 pm
Submission channel: a hyperlink on WBLE

Programme (IA/IB/CS/CN/CT):	**CS**
Group number:	**08**
Group leader name:	**Tan Jing Jie**

No.	**Name (in ascending order)**	**Student ID**	**UTAR email address**	**Practical Group**	**Signature****
1	Jacynth Tham Ming Quan	18ACB01600	jctmq25@1utar.my	P (2)	[illegible]
2	Tan Jing Jie	18ACB04560	tanjingjie@1utar.my	P (8)	Tan Jing Jie
3	Tan Wei Mun	18ACB03705	weimun19@1utar.my	P (2)	Wei
4	Yap Jheng Khin	18ACB00224	polarbearyap@1utar.my	P (19)	[illegible]

Note:
**All members should attach their individual signature confirming that the report is not plagiarized

Student ID As appeared on student card	**(Your ID)** **18ACB04560**	
Member name As appeared on student card	**(Your Name)** **Tan Jing Jie**	
Queries (30 marks)	**The following items shall be placed in this column:**	**Leave this column empty**

Transaction / question
Your SQL command/ your answers – as appear in your submitted *.sql
Partial / full OUTPUT screenshots

Query 1

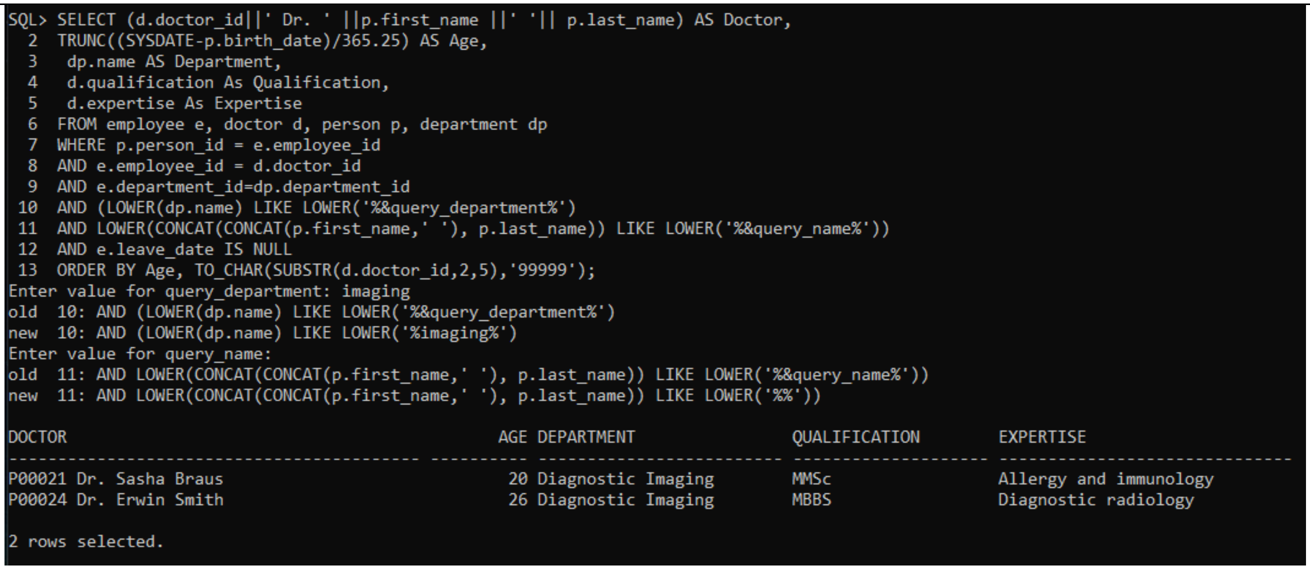
<u>Show the list of doctor(s) by using their name or/and department (prompting)</u>

- Receptionist or user can key in department name or/and doctor name (no case sensitive) to search about a list of related doctors in the hospital
- Receptionist or user can leave either department name as null to perform search by only doctor name, or vice versa doctor name as null to search only by doctor name.
- The is useful to help patient to filter out the doctor based on patient needs to perform treatment or service.
- This can be useful to find out related doctor expertise and his or her current department when patient query.

<u>SQL command</u>

```
SELECT (d.doctor_id||' Dr. ' ||p.first_name ||' '|| p.last_name) AS Doctor,
TRUNC((SYSDATE-p.birth_date)/365.25) AS Age,
 dp.name AS Department,
 d.qualification As Qualification,
 d.expertise As Expertise
FROM employee e, doctor d, person p, department dp
WHERE p.person_id = e.employee_id
AND e.employee_id = d.doctor_id
AND e.department_id=dp.department_id
AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
AND e.leave_date IS NULL
ORDER BY Age, TO_CHAR(SUBSTR(d.doctor_id,2,5),'99999');
```

<u>Screenshot</u>
Query department: imaging

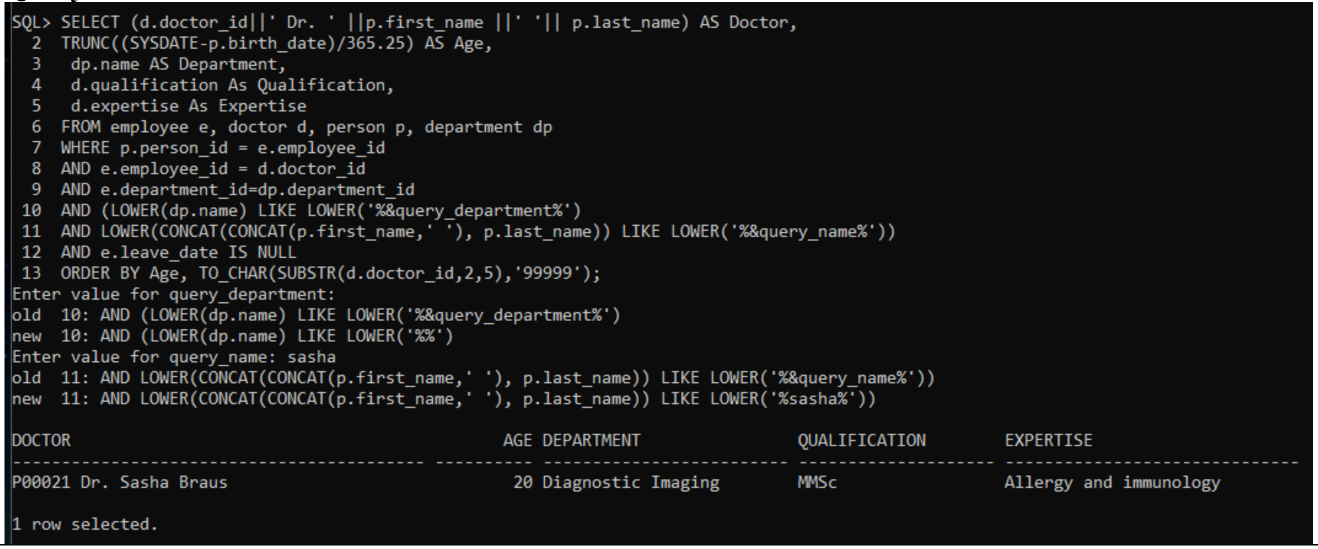


```
SQL> SELECT (d.doctor_id||' Dr. ' ||p.first_name ||' '|| p.last_name) AS Doctor,
  2  TRUNC((SYSDATE-p.birth_date)/365.25) AS Age,
  3   dp.name AS Department,
  4   d.qualification As Qualification,
  5   d.expertise As Expertise
  6  FROM employee e, doctor d, person p, department dp
  7  WHERE p.person_id = e.employee_id
  8  AND e.employee_id = d.doctor_id
  9  AND e.department_id=dp.department_id
 10  AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
 11  AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
 12  AND e.leave_date IS NULL
 13  ORDER BY Age, TO_CHAR(SUBSTR(d.doctor_id,2,5),'99999');
Enter value for query_department: imaging
old  10: AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
new  10: AND (LOWER(dp.name) LIKE LOWER('%imaging%')
Enter value for query_name:
old  11: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
new  11: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%%'))

DOCTOR                                          AGE DEPARTMENT               QUALIFICATION        EXPERTISE
------------------------------------------ ---------- ------------------------- -------------------- ------------------------------
P00021 Dr. Sasha Braus                           20 Diagnostic Imaging       MMSc                 Allergy and immunology
P00024 Dr. Erwin Smith                           26 Diagnostic Imaging       MBBS                 Diagnostic radiology

2 rows selected.
```

Query doctor name: sasha

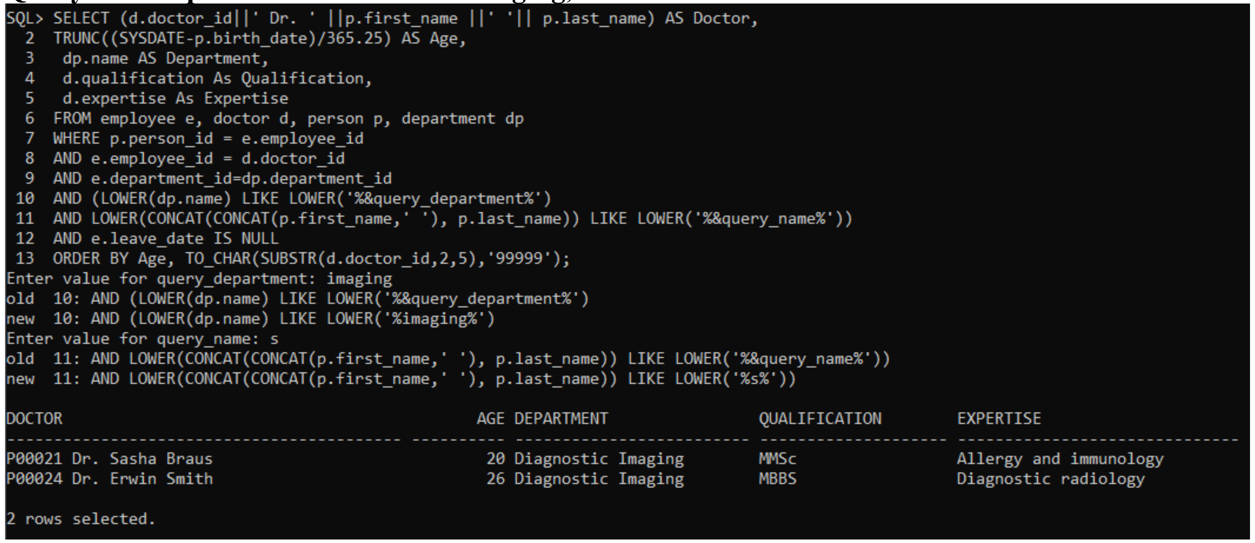


```
SQL> SELECT (d.doctor_id||' Dr. ' ||p.first_name ||' '|| p.last_name) AS Doctor,
  2  TRUNC((SYSDATE-p.birth_date)/365.25) AS Age,
  3   dp.name AS Department,
  4   d.qualification As Qualification,
  5   d.expertise As Expertise
  6  FROM employee e, doctor d, person p, department dp
  7  WHERE p.person_id = e.employee_id
  8  AND e.employee_id = d.doctor_id
  9  AND e.department_id=dp.department_id
 10  AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
 11  AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
 12  AND e.leave_date IS NULL
 13  ORDER BY Age, TO_CHAR(SUBSTR(d.doctor_id,2,5),'99999');
Enter value for query_department:
old  10: AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
new  10: AND (LOWER(dp.name) LIKE LOWER('%%')
Enter value for query_name: sasha
old  11: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
new  11: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%sasha%'))

DOCTOR                                          AGE DEPARTMENT               QUALIFICATION        EXPERTISE
------------------------------------------ ---------- ------------------------- -------------------- ------------------------------
P00021 Dr. Sasha Braus                           20 Diagnostic Imaging       MMSc                 Allergy and immunology

1 row selected.
```

Query both department and doctor: imaging, s

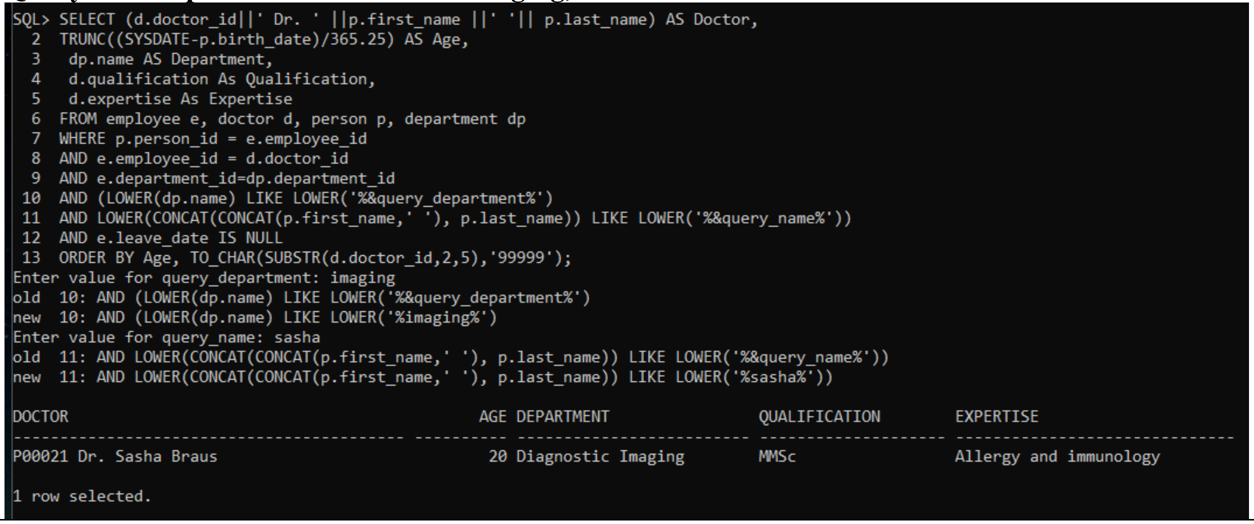


```
SQL> SELECT (d.doctor_id||' Dr. ' ||p.first_name ||' '|| p.last_name) AS Doctor,
  2  TRUNC((SYSDATE-p.birth_date)/365.25) AS Age,
  3   dp.name AS Department,
  4   d.qualification As Qualification,
  5   d.expertise As Expertise
  6  FROM employee e, doctor d, person p, department dp
  7  WHERE p.person_id = e.employee_id
  8  AND e.employee_id = d.doctor_id
  9  AND e.department_id=dp.department_id
 10  AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
 11  AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
 12  AND e.leave_date IS NULL
 13  ORDER BY Age, TO_CHAR(SUBSTR(d.doctor_id,2,5),'99999');
Enter value for query_department: imaging
old  10: AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
new  10: AND (LOWER(dp.name) LIKE LOWER('%imaging%')
Enter value for query_name: s
old  11: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
new  11: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%s%'))

DOCTOR                                          AGE DEPARTMENT               QUALIFICATION        EXPERTISE
---------------------------------------- ---------- ------------------------ -------------------- ------------------------------
P00021 Dr. Sasha Braus                           20 Diagnostic Imaging       MMSc                 Allergy and immunology
P00024 Dr. Erwin Smith                           26 Diagnostic Imaging       MBBS                 Diagnostic radiology

2 rows selected.
```

Query both department and doctor: imaging, sasha



```
SQL> SELECT (d.doctor_id||' Dr. ' ||p.first_name ||' '|| p.last_name) AS Doctor,
  2  TRUNC((SYSDATE-p.birth_date)/365.25) AS Age,
  3   dp.name AS Department,
  4   d.qualification As Qualification,
  5   d.expertise As Expertise
  6  FROM employee e, doctor d, person p, department dp
  7  WHERE p.person_id = e.employee_id
  8  AND e.employee_id = d.doctor_id
  9  AND e.department_id=dp.department_id
 10  AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
 11  AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
 12  AND e.leave_date IS NULL
 13  ORDER BY Age, TO_CHAR(SUBSTR(d.doctor_id,2,5),'99999');
Enter value for query_department: imaging
old  10: AND (LOWER(dp.name) LIKE LOWER('%&query_department%')
new  10: AND (LOWER(dp.name) LIKE LOWER('%imaging%')
Enter value for query_name: sasha
old  11: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&query_name%'))
new  11: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%sasha%'))

DOCTOR                                          AGE DEPARTMENT               QUALIFICATION        EXPERTISE
---------------------------------------- ---------- ------------------------ -------------------- ------------------------------
P00021 Dr. Sasha Braus                           20 Diagnostic Imaging       MMSc                 Allergy and immunology

1 row selected.
```

Query 2

<u>Show all service that perform to patient in an admission (by prompting patient name)</u>

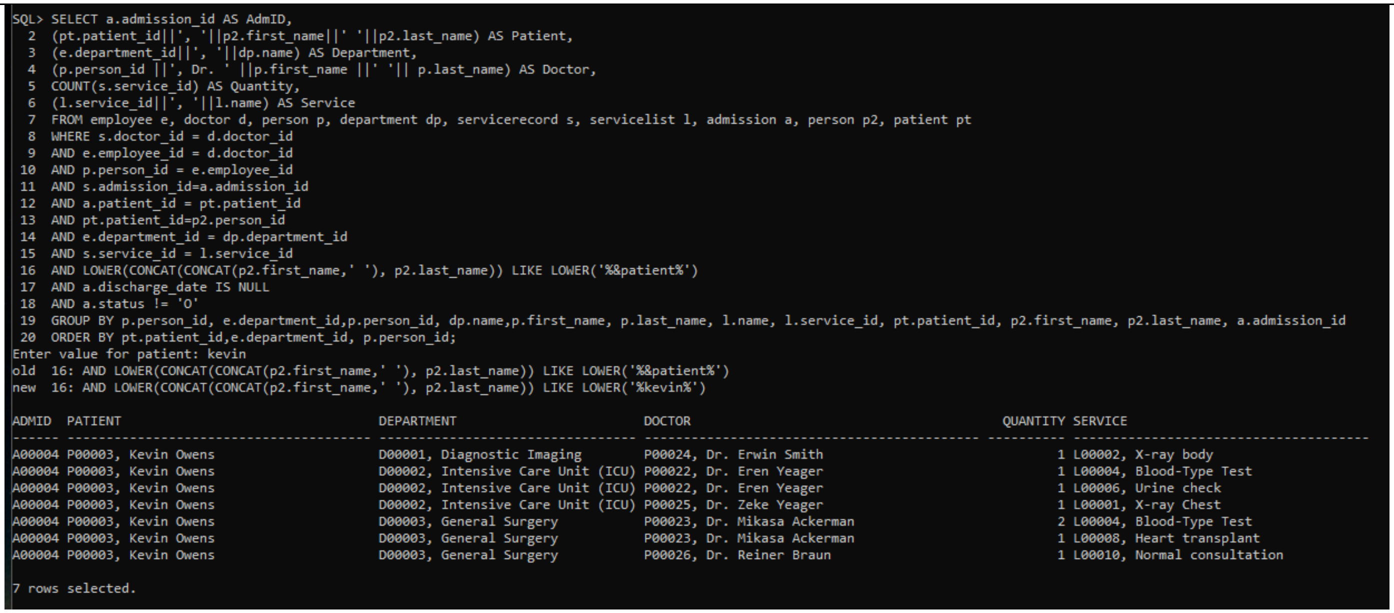
- Query can done by input of patient name (first name or last name or part of name will do and no case sensitive)
- This can be useful to find out all the doctor and services perform to a patient in one admission
- This is useful when a group of doctors want to have a discussion meeting before they do surgery operation to a patient.

<u>SQL command</u>

```
SELECT a.admission_id AS AdmID,
(pt.patient_id||', '||p2.first_name||' '||p2.last_name) AS Patient,
(e.department_id||', '||dp.name) AS Department,
(p.person_id ||', Dr. ' ||p.first_name ||' '|| p.last_name) AS Doctor,
COUNT(s.service_id) AS Quantity,
(l.service_id||', '||l.name) AS Service
FROM employee e, doctor d, person p, department dp, servicerecord s, servicelist l, admission a, person p2, patient pt
WHERE s.doctor_id = d.doctor_id
AND e.employee_id = d.doctor_id
AND p.person_id = e.employee_id
AND s.admission_id=a.admission_id
AND a.patient_id = pt.patient_id
AND pt.patient_id=p2.person_id
AND e.department_id = dp.department_id
AND s.service_id = l.service_id
AND LOWER(CONCAT(CONCAT(p2.first_name,' '), p2.last_name)) LIKE LOWER('%&patient%')
AND a.discharge_date IS NULL
AND a.status != 'O'
GROUP BY p.person_id, e.department_id,p.person_id, dp.name,p.first_name, p.last_name, l.name, l.service_id,
pt.patient_id, p2.first_name, p2.last_name, a.admission_id
ORDER BY pt.patient_id,e.department_id, p.person_id;
```

<u>Screenshot</u>

Query by patient name: kevin

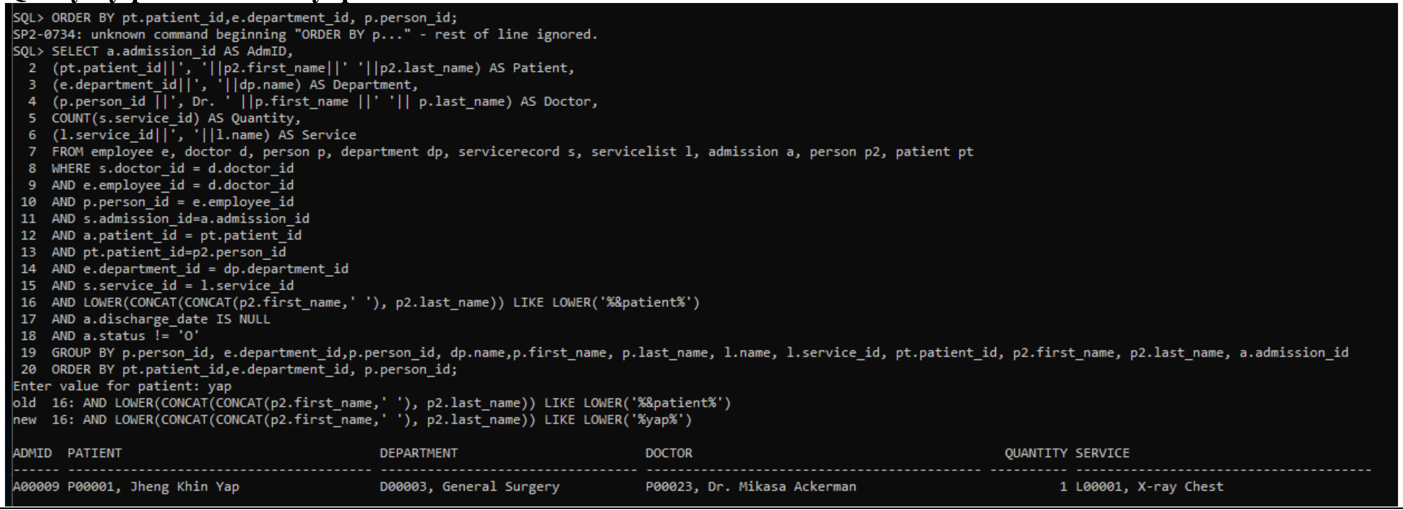


```
SQL> SELECT a.admission_id AS AdmID,
  2  (pt.patient_id||', '||p2.first_name||' '||p2.last_name) AS Patient,
  3  (e.department_id||', '||dp.name) AS Department,
  4  (p.person_id ||', Dr. ' ||p.first_name ||' '|| p.last_name) AS Doctor,
  5  COUNT(s.service_id) AS Quantity,
  6  (l.service_id||', '||l.name) AS Service
  7  FROM employee e, doctor d, person p, department dp, servicerecord s, servicelist l, admission a, person p2, patient pt
  8  WHERE s.doctor_id = d.doctor_id
  9  AND e.employee_id = d.doctor_id
 10  AND p.person_id = e.employee_id
 11  AND s.admission_id=a.admission_id
 12  AND a.patient_id = pt.patient_id
 13  AND pt.patient_id=p2.person_id
 14  AND e.department_id = dp.department_id
 15  AND s.service_id = l.service_id
 16  AND LOWER(CONCAT(CONCAT(p2.first_name,' '), p2.last_name)) LIKE LOWER('%&patient%')
 17  AND a.discharge_date IS NULL
 18  AND a.status != 'O'
 19  GROUP BY p.person_id, e.department_id,p.person_id, dp.name,p.first_name, p.last_name, l.name, l.service_id, pt.patient_id, p2.first_name, p2.last_name, a.admission_id
 20  ORDER BY pt.patient_id,e.department_id, p.person_id;
Enter value for patient: kevin
old  16: AND LOWER(CONCAT(CONCAT(p2.first_name,' '), p2.last_name)) LIKE LOWER('%&patient%')
new  16: AND LOWER(CONCAT(CONCAT(p2.first_name,' '), p2.last_name)) LIKE LOWER('%kevin%')

ADMID  PATIENT                                  DEPARTMENT                         DOCTOR                                           QUANTITY SERVICE
------ ---------------------------------------- ---------------------------------- ------------------------------------------------ --------- ----------------------------------------
A00004 P00003, Kevin Owens                      D00001, Diagnostic Imaging         P00024, Dr. Erwin Smith                                  1 L00002, X-ray body
A00004 P00003, Kevin Owens                      D00002, Intensive Care Unit (ICU) P00022, Dr. Eren Yeager                                  1 L00004, Blood-Type Test
A00004 P00003, Kevin Owens                      D00002, Intensive Care Unit (ICU) P00022, Dr. Eren Yeager                                  1 L00006, Urine check
A00004 P00003, Kevin Owens                      D00002, Intensive Care Unit (ICU) P00025, Dr. Zeke Yeager                                  1 L00001, X-ray Chest
A00004 P00003, Kevin Owens                      D00003, General Surgery            P00023, Dr. Mikasa Ackerman                              2 L00004, Blood-Type Test
A00004 P00003, Kevin Owens                      D00003, General Surgery            P00023, Dr. Mikasa Ackerman                              1 L00008, Heart transplant
A00004 P00003, Kevin Owens                      D00003, General Surgery            P00026, Dr. Reiner Braun                                 1 L00010, Normal consultation

7 rows selected.
```

Query by patient name: yap



```
SQL> ORDER BY pt.patient_id,e.department_id, p.person_id;
SP2-0734: unknown command beginning "ORDER BY p..." - rest of line ignored.
SQL> SELECT a.admission_id AS AdmID,
  2  (pt.patient_id||', '||p2.first_name||' '||p2.last_name) AS Patient,
  3  (e.department_id||', '||dp.name) AS Department,
  4  (p.person_id ||', Dr. ' ||p.first_name ||' '|| p.last_name) AS Doctor,
  5  COUNT(s.service_id) AS Quantity,
  6  (l.service_id||', '||l.name) AS Service
  7  FROM employee e, doctor d, person p, department dp, servicerecord s, servicelist l, admission a, person p2, patient pt
  8  WHERE s.doctor_id = d.doctor_id
  9  AND e.employee_id = d.doctor_id
 10  AND p.person_id = e.employee_id
 11  AND s.admission_id=a.admission_id
 12  AND a.patient_id = pt.patient_id
 13  AND pt.patient_id=p2.person_id
 14  AND e.department_id = dp.department_id
 15  AND s.service_id = l.service_id
 16  AND LOWER(CONCAT(CONCAT(p2.first_name,' '), p2.last_name)) LIKE LOWER('%&patient%')
 17  AND a.discharge_date IS NULL
 18  AND a.status != 'O'
 19  GROUP BY p.person_id, e.department_id,p.person_id, dp.name,p.first_name, p.last_name, l.name, l.service_id, pt.patient_id, p2.first_name, p2.last_name, a.admission_id
 20  ORDER BY pt.patient_id,e.department_id, p.person_id;
Enter value for patient: yap
old  16: AND LOWER(CONCAT(CONCAT(p2.first_name,' '), p2.last_name)) LIKE LOWER('%&patient%')
new  16: AND LOWER(CONCAT(CONCAT(p2.first_name,' '), p2.last_name)) LIKE LOWER('%yap%')

ADMID  PATIENT                                  DEPARTMENT                         DOCTOR                                           QUANTITY SERVICE
------ ---------------------------------------- ---------------------------------- ------------------------------------------------ --------- ----------------------------------------
A00009 P00001, Jheng Khin Yap                   D00003, General Surgery            P00023, Dr. Mikasa Ackerman                              1 L00001, X-ray Chest
```

Query 3

Show a list of a doctor's patients (by prompting doctor ID)

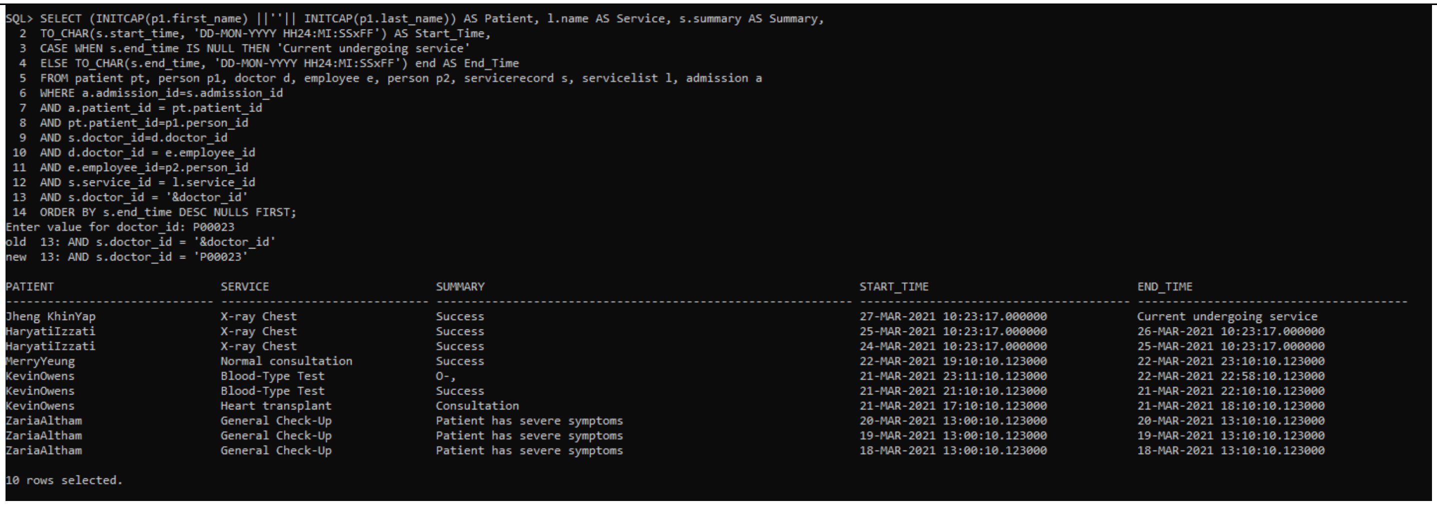
- Query can do by using a doctor's ID
- This can be useful to find out all the patients for a doctor that he or she performed service before
- This is useful when doctor want to summaries what they do their patient for their report writing.
- This query will list out according to the time schedule (the latest first).
- This query is helpful when a doctor wants to trace back list of patient history.
- This query can also check a doctor whether he or she is free from provide service to patient.
- This query is useful when a doctor contracted Covid-19 and hospital want to trace back who visited the doctor before.
- If pt.patient_id added in Order By clause (first position), then it will arrange according to patient then continue with its chronological order.

SQL command

```
SELECT (INITCAP(p1.first_name) ||''|| INITCAP(p1.last_name)) AS Patient, l.name AS Service, s.summary AS
Summary,
TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI:SSxFF') AS Start_Time,
CASE WHEN s.end_time IS NULL THEN 'Current undergoing service'
    ELSE TO_CHAR(s.end_time, 'DD-MON-YYYY HH24:MI:SSxFF') end AS End_Time
FROM patient pt, person p1, doctor d, employee e, person p2, servicerecord s, servicelist l, admission a
WHERE a.admission_id=s.admission_id
AND a.patient_id = pt.patient_id
AND pt.patient_id=p1.person_id
AND s.doctor_id=d.doctor_id
AND d.doctor_id = e.employee_id
AND e.employee_id=p2.person_id
AND s.service_id = l.service_id
AND s.doctor_id = '&doctor_id'
ORDER BY s.end_time DESC NULLS FIRST;
```

Screenshot

Query by doctor id : P00023

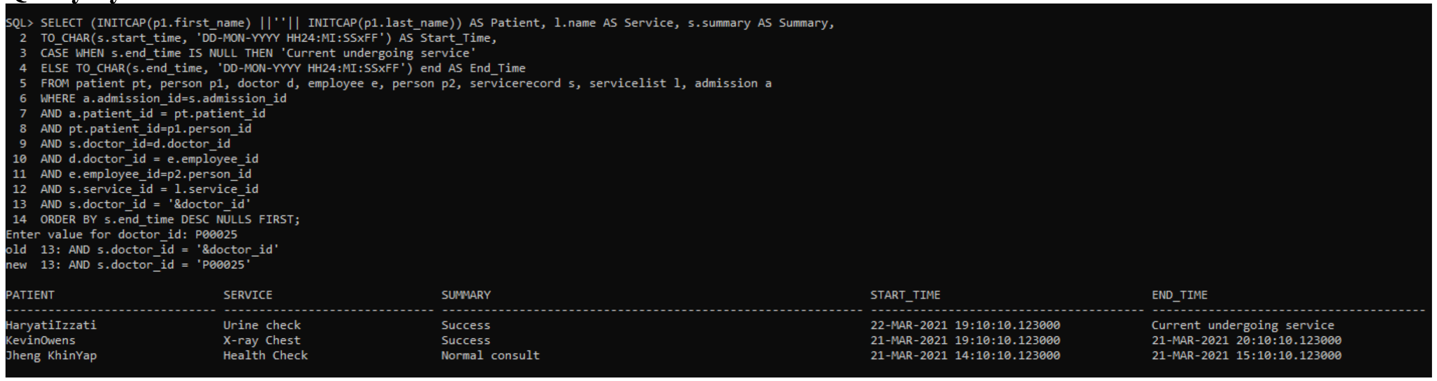


```
SQL> SELECT (INITCAP(p1.first_name) ||''|| INITCAP(p1.last_name)) AS Patient, l.name AS Service, s.summary AS Summary,
  2  TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI:SSxFF') AS Start_Time,
  3  CASE WHEN s.end_time IS NULL THEN 'Current undergoing service'
  4  ELSE TO_CHAR(s.end_time, 'DD-MON-YYYY HH24:MI:SSxFF') end AS End_Time
  5  FROM patient pt, person p1, doctor d, employee e, person p2, servicerecord s, servicelist l, admission a
  6  WHERE a.admission_id=s.admission_id
  7  AND a.patient_id = pt.patient_id
  8  AND pt.patient_id=p1.person_id
  9  AND s.doctor_id=d.doctor_id
 10  AND d.doctor_id = e.employee_id
 11  AND e.employee_id=p2.person_id
 12  AND s.service_id = l.service_id
 13  AND s.doctor_id = '&doctor_id'
 14  ORDER BY s.end_time DESC NULLS FIRST;
Enter value for doctor_id: P00023
old  13: AND s.doctor_id = '&doctor_id'
new  13: AND s.doctor_id = 'P00023'

PATIENT                        SERVICE                        SUMMARY                                                        START_TIME                                 END_TIME
------------------------------ ------------------------------ -------------------------------------------------------------- ------------------------------------------ ---------------------------------------
Jheng KhinYap                  X-ray Chest                    Success                                                        27-MAR-2021 10:23:17.000000                Current undergoing service
HaryatiIzzati                  X-ray Chest                    Success                                                        25-MAR-2021 10:23:17.000000                26-MAR-2021 10:23:17.000000
HaryatiIzzati                  X-ray Chest                    Success                                                        24-MAR-2021 10:23:17.000000                25-MAR-2021 10:23:17.000000
MerryYeung                     Normal consultation            Success                                                        22-MAR-2021 19:10:10.123000                22-MAR-2021 23:10:10.123000
KevinOwens                     Blood-Type Test                O-,                                                            21-MAR-2021 23:11:10.123000                22-MAR-2021 22:58:10.123000
KevinOwens                     Blood-Type Test                Success                                                        21-MAR-2021 21:10:10.123000                21-MAR-2021 22:10:10.123000
KevinOwens                     Heart transplant               Consultation                                                   21-MAR-2021 17:10:10.123000                21-MAR-2021 18:10:10.123000
ZariaAltham                    General Check-Up               Patient has severe symptoms                                    20-MAR-2021 13:00:10.123000                20-MAR-2021 13:10:10.123000
ZariaAltham                    General Check-Up               Patient has severe symptoms                                    19-MAR-2021 13:00:10.123000                19-MAR-2021 13:10:10.123000
ZariaAltham                    General Check-Up               Patient has severe symptoms                                    18-MAR-2021 13:00:10.123000                18-MAR-2021 13:10:10.123000

10 rows selected.
```

Query by doctor id : P00025



```
SQL> SELECT (INITCAP(p1.first_name) ||''|| INITCAP(p1.last_name)) AS Patient, l.name AS Service, s.summary AS Summary,
  2  TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI:SSxFF') AS Start_Time,
  3  CASE WHEN s.end_time IS NULL THEN 'Current undergoing service'
  4  ELSE TO_CHAR(s.end_time, 'DD-MON-YYYY HH24:MI:SSxFF') end AS End_Time
  5  FROM patient pt, person p1, doctor d, employee e, person p2, servicerecord s, servicelist l, admission a
  6  WHERE a.admission_id=s.admission_id
  7  AND a.patient_id = pt.patient_id
  8  AND pt.patient_id=p1.person_id
  9  AND s.doctor_id=d.doctor_id
 10  AND d.doctor_id = e.employee_id
 11  AND e.employee_id=p2.person_id
 12  AND s.service_id = l.service_id
 13  AND s.doctor_id = '&doctor_id'
 14  ORDER BY s.end_time DESC NULLS FIRST;
Enter value for doctor_id: P00025
old  13: AND s.doctor_id = '&doctor_id'
new  13: AND s.doctor_id = 'P00025'

PATIENT                        SERVICE                        SUMMARY                                                        START_TIME                                 END_TIME
------------------------------ ------------------------------ -------------------------------------------------------------- ------------------------------------------ ---------------------------------------
HaryatiIzzati                  Urine check                    Success                                                        22-MAR-2021 19:10:10.123000                Current undergoing service
KevinOwens                     X-ray Chest                    Success                                                        21-MAR-2021 19:10:10.123000                21-MAR-2021 20:10:10.123000
Jheng KhinYap                  Health Check                   Normal consult                                                 21-MAR-2021 14:10:10.123000                21-MAR-2021 15:10:10.123000
```

Query 4

<u>Show list of unpaid including patient contact after due date</u>

- This query will list out related patient contact when he or she have overdue bill.
- This hospital staff can use this query to sort out the details list of unpaid bill and contact the patients for payment procedure.
- This query can be slightly modified in order to sort out the list for future. For instance, when change the SYSDATE to tomorrow date, then this query can query for tomorrow due bill, so that staff can notify the patient before incur penalty

charge to them. In order to change SYSDATE to tomorrow we can use TO_DATE(’27-03-21’,’DD-MM-YY’) or instead SYSDATE+1.

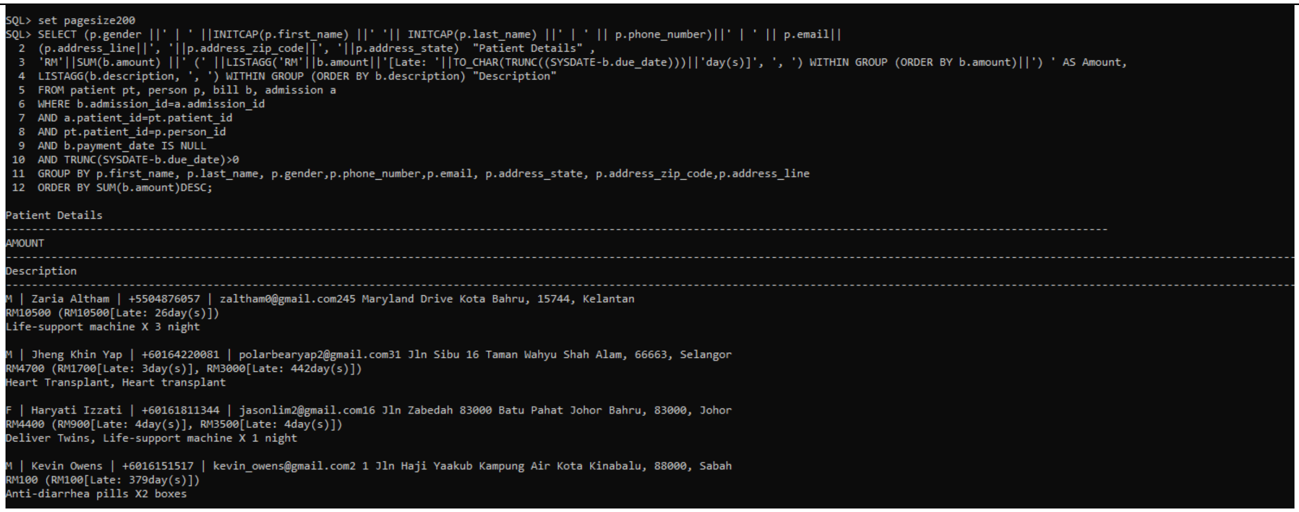
- Hence obviously, when twisting to query list of bill due yesterday or the day after tomorrow can be done by put ‘-1’ and ‘+2’ respectively.

SQL command

```
SELECT (p.gender ||' | ' ||INITCAP(p.first_name) ||' '|| INITCAP(p.last_name) ||' | ' || p.phone_number)||' | ' || p.email||
(p.address_line||', '||p.address_zip_code||', '||p.address_state)  "Patient Details" ,
'RM'||SUM(b.amount) ||' (' ||LISTAGG('RM'||b.amount||'[Late: '||TO_CHAR(TRUNC((SYSDATE-b.due_date)))||'day(s)]',
', ') WITHIN GROUP (ORDER BY b.amount)||') ' AS Amount,
LISTAGG(b.description, ', ') WITHIN GROUP (ORDER BY b.description) "Description"
FROM patient pt, person p, bill b, admission a
WHERE b.admission_id=a.admission_id
AND a.patient_id=pt.patient_id
AND pt.patient_id=p.person_id
AND b.payment_date IS NULL
AND TRUNC(SYSDATE-b.due_date)>0
GROUP BY p.first_name, p.last_name, p.gender,p.phone_number,p.email, p.address_state,
p.address_zip_code,p.address_line
ORDER BY SUM(b.amount)DESC;
```

Screenshot

Using SYSDATE: list the bill expired today



```
SQL> set pagesize200
SQL> SELECT (p.gender ||' | ' ||INITCAP(p.first_name) ||' '|| INITCAP(p.last_name) ||' | ' || p.phone_number)||' | ' || p.email||
  2  (p.address_line||', '||p.address_zip_code||', '||p.address_state)  "Patient Details" ,
  3  'RM'||SUM(b.amount) ||' (' ||LISTAGG('RM'||b.amount||'[Late: '||TO_CHAR(TRUNC((SYSDATE-b.due_date)))||'day(s)]', ', ') WITHIN GROUP (ORDER BY b.amount)||') ' AS Amount,
  4  LISTAGG(b.description, ', ') WITHIN GROUP (ORDER BY b.description) "Description"
  5  FROM patient pt, person p, bill b, admission a
  6  WHERE b.admission_id=a.admission_id
  7  AND a.patient_id=pt.patient_id
  8  AND pt.patient_id=p.person_id
  9  AND b.payment_date IS NULL
 10  AND TRUNC(SYSDATE-b.due_date)>0
 11  GROUP BY p.first_name, p.last_name, p.gender,p.phone_number,p.email, p.address_state, p.address_zip_code,p.address_line
 12  ORDER BY SUM(b.amount)DESC;

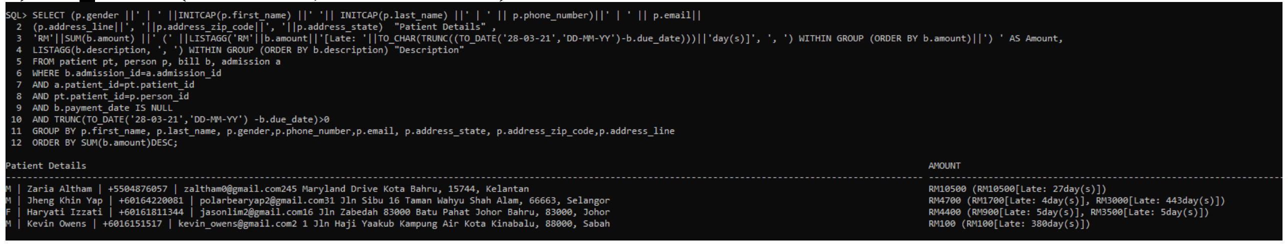
Patient Details
--------------------------------------------------------------------------------------------------------------------------------------------------
AMOUNT
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------
Description
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------
M | Zaria Altham | +5504876057 | zaltham0@gmail.com245 Maryland Drive Kota Bahru, 15744, Kelantan
RM10500 (RM10500[Late: 26day(s)])
Life-support machine X 3 night

M | Jheng Khin Yap | +60164220081 | polarbearyap2@gmail.com31 Jln Sibu 16 Taman Wahyu Shah Alam, 66663, Selangor
RM4700 (RM1700[Late: 3day(s)], RM3000[Late: 442day(s)])
Heart Transplant, Heart transplant

F | Haryati Izzati | +60161811344 | jasonlim2@gmail.com16 Jln Zabedah 83000 Batu Pahat Johor Bahru, 83000, Johor
RM4400 (RM900[Late: 4day(s)], RM3500[Late: 4day(s)])
Deliver Twins, Life-support machine X 1 night

M | Kevin Owens | +6016151517 | kevin_owens@gmail.com2 1 Jln Haji Yaakub Kampung Air Kota Kinabalu, 88000, Sabah
RM100 (RM100[Late: 379day(s)])
Anti-diarrhea pills X2 boxes
```

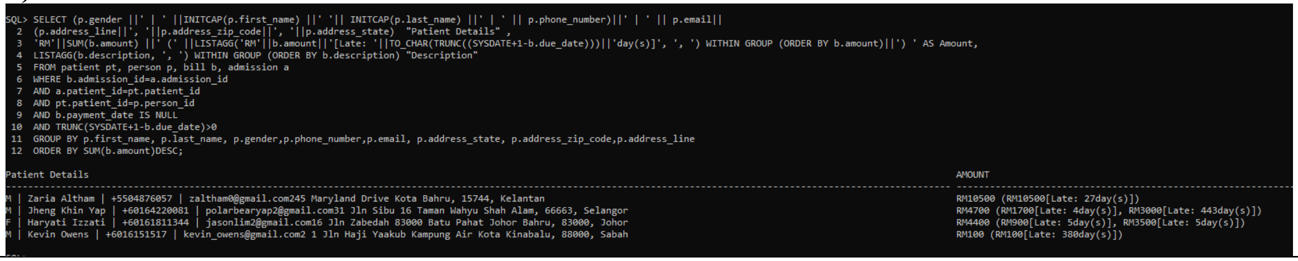
Using TO_DATE('27-03-21','DD-MM-YY') or SYSDATE+1: list the bill expired tomorrow. (Alter the script)
Both show the same result
a) TO_DATE('27-03-21','DD-MM-YY')



```
SQL> SELECT (p.gender ||' | ' ||INITCAP(p.first_name) ||' '|| INITCAP(p.last_name) ||' | ' || p.phone_number)||' | ' || p.email||
  2  (p.address_line||', '||p.address_zip_code||', '||p.address_state)  "Patient Details" ,
  3  'RM'||SUM(b.amount) ||' (' ||LISTAGG('RM'||b.amount||'[Late: '||TO_CHAR(TRUNC((TO_DATE('28-03-21','DD-MM-YY')-b.due_date)))||'day(s)]', ', ') WITHIN GROUP (ORDER BY b.amount)||') ' AS Amount,
  4  LISTAGG(b.description, ', ') WITHIN GROUP (ORDER BY b.description) "Description"
  5  FROM patient pt, person p, bill b, admission a
  6  WHERE b.admission_id=a.admission_id
  7  AND a.patient_id=pt.patient_id
  8  AND pt.patient_id=p.person_id
  9  AND b.payment_date IS NULL
 10  AND TRUNC(TO_DATE('28-03-21','DD-MM-YY') -b.due_date)>0
 11  GROUP BY p.first_name, p.last_name, p.gender,p.phone_number,p.email, p.address_state, p.address_zip_code,p.address_line
 12  ORDER BY SUM(b.amount)DESC;

Patient Details                                                                                                            AMOUNT
-------------------------------------------------------------------------------------------------------------------------- ----------------------------------------------
M | Zaria Altham | +5504876057 | zaltham0@gmail.com245 Maryland Drive Kota Bahru, 15744, Kelantan                            RM10500 (RM10500[Late: 27day(s)])
M | Jheng Khin Yap | +60164220081 | polarbearyap2@gmail.com31 Jln Sibu 16 Taman Wahyu Shah Alam, 66663, Selangor             RM4700 (RM1700[Late: 4day(s)], RM3000[Late: 443day(s)])
F | Haryati Izzati | +60161811344 | jasonlim2@gmail.com16 Jln Zabedah 83000 Batu Pahat Johor Bahru, 83000, Johor             RM4400 (RM900[Late: 5day(s)], RM3500[Late: 5day(s)])
M | Kevin Owens | +6016151517 | kevin_owens@gmail.com2 1 Jln Haji Yaakub Kampung Air Kota Kinabalu, 88000, Sabah             RM100 (RM100[Late: 380day(s)])
```

b) SYSDATE+1:



```
SQL> SELECT (p.gender ||' | ' ||INITCAP(p.first_name) ||' '|| INITCAP(p.last_name) ||' | ' || p.phone_number)||' | ' || p.email||
  2  (p.address_line||', '||p.address_zip_code||', '||p.address_state)  "Patient Details" ,
  3  'RM'||SUM(b.amount) ||' (' ||LISTAGG('RM'||b.amount||'[Late: '||TO_CHAR(TRUNC((SYSDATE+1-b.due_date)))||'day(s)]', ', ') WITHIN GROUP (ORDER BY b.amount)||') ' AS Amount,
  4  LISTAGG(b.description, ', ') WITHIN GROUP (ORDER BY b.description) "Description"
  5  FROM patient pt, person p, bill b, admission a
  6  WHERE b.admission_id=a.admission_id
  7  AND a.patient_id=pt.patient_id
  8  AND pt.patient_id=p.person_id
  9  AND b.payment_date IS NULL
 10  AND TRUNC(SYSDATE+1-b.due_date)>0
 11  GROUP BY p.first_name, p.last_name, p.gender,p.phone_number,p.email, p.address_state, p.address_zip_code,p.address_line
 12  ORDER BY SUM(b.amount)DESC;

Patient Details                                                                                                            AMOUNT
-------------------------------------------------------------------------------------------------------------------------- ----------------------------------------------
M | Zaria Altham | +5504876057 | zaltham0@gmail.com245 Maryland Drive Kota Bahru, 15744, Kelantan                            RM10500 (RM10500[Late: 27day(s)])
M | Jheng Khin Yap | +60164220081 | polarbearyap2@gmail.com31 Jln Sibu 16 Taman Wahyu Shah Alam, 66663, Selangor             RM4700 (RM1700[Late: 4day(s)], RM3000[Late: 443day(s)])
F | Haryati Izzati | +60161811344 | jasonlim2@gmail.com16 Jln Zabedah 83000 Batu Pahat Johor Bahru, 83000, Johor             RM4400 (RM900[Late: 5day(s)], RM3500[Late: 5day(s)])
M | Kevin Owens | +6016151517 | kevin_owens@gmail.com2 1 Jln Haji Yaakub Kampung Air Kota Kinabalu, 88000, Sabah             RM100 (RM100[Late: 380day(s)])
```

Query 5	**Show list admission which stay in hospital from current date. (prompting the nearest n day(s))**	

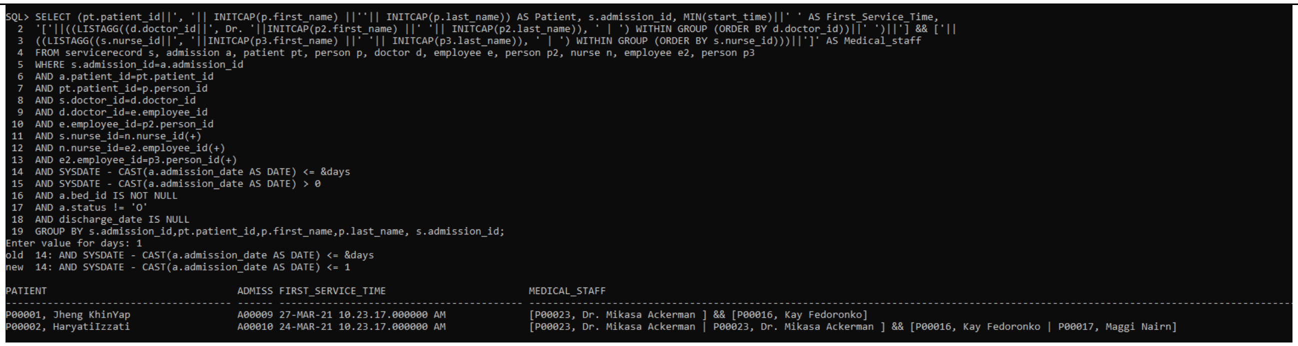
- This query will list out all admission from nearest n days.
- This query prompt user to input the nearest n days
- This query is useful when to print out a check list for preparing food, normal patrol (This can be achieve by setting the nearest day to a very huge number, for instance 99999999), and do analysis today admission patient. It can show out how many patients come in today or total up from previous days.

<u>SQL command</u>

```
SELECT (pt.patient_id||', '|| INITCAP(p.first_name) ||''|| INITCAP(p.last_name)) AS Patient, s.admission_id,
MIN(start_time)||' ' AS First_Service_Time,
'['||((LISTAGG((d.doctor_id||', Dr. '||INITCAP(p2.first_name) ||' '|| INITCAP(p2.last_name)), ' | ') WITHIN GROUP
(ORDER BY d.doctor_id))||' ')||'] && ['||
((LISTAGG((s.nurse_id||', '||INITCAP(p3.first_name) ||' '|| INITCAP(p3.last_name)), ' | ') WITHIN GROUP (ORDER
BY s.nurse_id)))||']' AS Medical_staff
FROM servicerecord s, admission a, patient pt, person p, doctor d, employee e, person p2, nurse n, employee e2, person
p3
WHERE s.admission_id=a.admission_id
AND a.patient_id=pt.patient_id
AND pt.patient_id=p.person_id
AND s.doctor_id=d.doctor_id
AND d.doctor_id=e.employee_id
AND e.employee_id=p2.person_id
AND s.nurse_id=n.nurse_id(+)
AND n.nurse_id=e2.employee_id(+)
AND e2.employee_id=p3.person_id(+)
AND SYSDATE - CAST(a.admission_date AS DATE) <= &days
AND SYSDATE - CAST(a.admission_date AS DATE) > 0
AND a.bed_id IS NOT NULL
AND a.status != 'O'
AND discharge_date IS NULL
GROUP BY s.admission_id,pt.patient_id,p.first_name,p.last_name, s.admission_id;
```

<u>Screenshot</u>

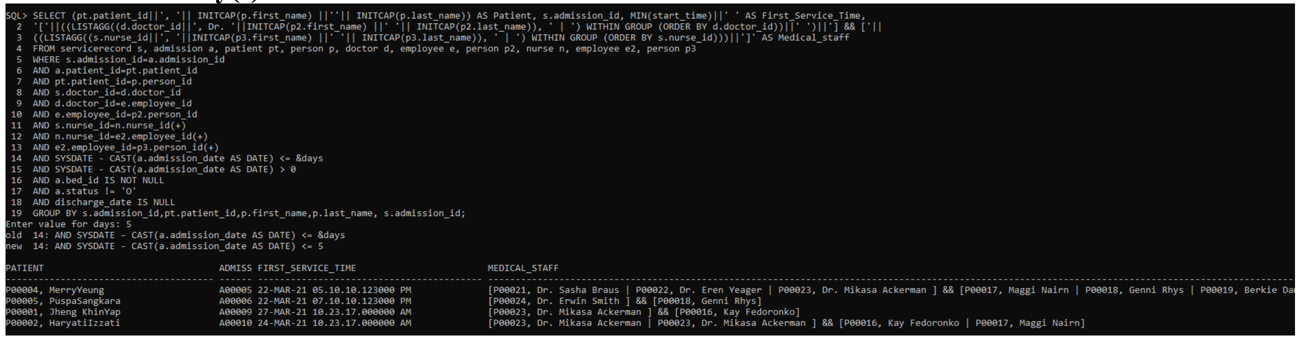
The nearest n day(s) : 1



```
SQL> SELECT (pt.patient_id||', '|| INITCAP(p.first_name) ||''|| INITCAP(p.last_name)) AS Patient, s.admission_id, MIN(start_time)||' ' AS First_Service_Time,
  2  '['||((LISTAGG((d.doctor_id||', Dr. '||INITCAP(p2.first_name) ||' '|| INITCAP(p2.last_name)), ' | ') WITHIN GROUP (ORDER BY d.doctor_id))||' ')||'] && ['||
  3  ((LISTAGG((s.nurse_id||', '||INITCAP(p3.first_name) ||' '|| INITCAP(p3.last_name)), ' | ') WITHIN GROUP (ORDER BY s.nurse_id)))||']' AS Medical_staff
  4  FROM servicerecord s, admission a, patient pt, person p, doctor d, employee e, person p2, nurse n, employee e2, person p3
  5  WHERE s.admission_id=a.admission_id
  6  AND a.patient_id=pt.patient_id
  7  AND pt.patient_id=p.person_id
  8  AND s.doctor_id=d.doctor_id
  9  AND d.doctor_id=e.employee_id
 10  AND e.employee_id=p2.person_id
 11  AND s.nurse_id=n.nurse_id(+)
 12  AND n.nurse_id=e2.employee_id(+)
 13  AND e2.employee_id=p3.person_id(+)
 14  AND SYSDATE - CAST(a.admission_date AS DATE) <= &days
 15  AND SYSDATE - CAST(a.admission_date AS DATE) > 0
 16  AND a.bed_id IS NOT NULL
 17  AND a.status != 'O'
 18  AND discharge_date IS NULL
 19  GROUP BY s.admission_id,pt.patient_id,p.first_name,p.last_name, s.admission_id;
Enter value for days: 1
old  14: AND SYSDATE - CAST(a.admission_date AS DATE) <= &days
new  14: AND SYSDATE - CAST(a.admission_date AS DATE) <= 1

PATIENT                                 ADMISS FIRST_SERVICE_TIME                          MEDICAL_STAFF
--------------------------------------- ------ ------------------------------------------- ------------------------------------------------------------------------------------------------------------------------------------
P00001, Jheng KhinYap                   A00009 27-MAR-21 10.23.17.000000 AM                [P00023, Dr. Mikasa Ackerman ] && [P00016, Kay Fedoronko]
P00002, HaryatiIzzati                   A00010 24-MAR-21 10.23.17.000000 AM                [P00023, Dr. Mikasa Ackerman | P00023, Dr. Mikasa Ackerman ] && [P00016, Kay Fedoronko | P00017, Maggi Nairn]
```

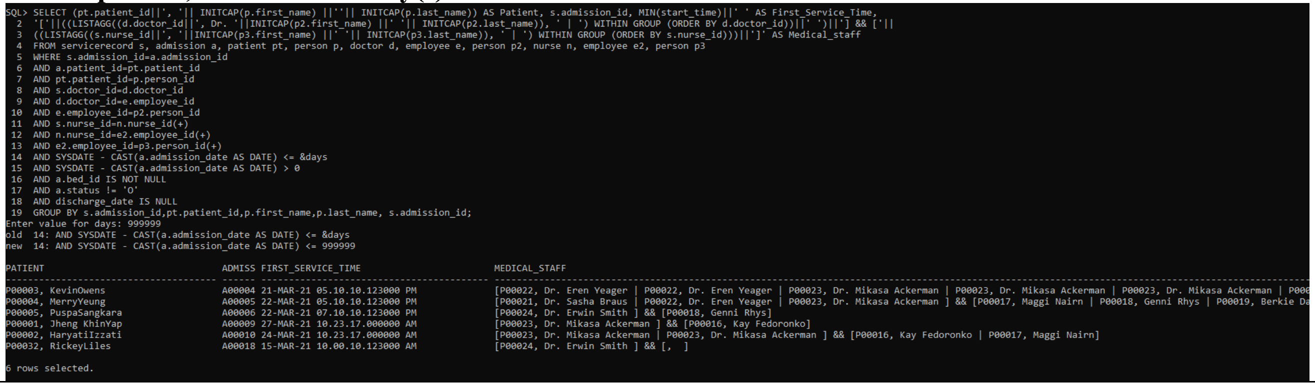
The nearest n day(s):5



```
SQL> SELECT (pt.patient_id||', '|| INITCAP(p.first_name) ||''|| INITCAP(p.last_name)) AS Patient, s.admission_id, MIN(start_time)||' ' AS First_Service_Time,
  2  '['||((LISTAGG((d.doctor_id||', Dr. '||INITCAP(p2.first_name) ||' '|| INITCAP(p2.last_name)), ' | ') WITHIN GROUP (ORDER BY d.doctor_id))||' ')||'] && ['||
  3  ((LISTAGG((s.nurse_id||', '||INITCAP(p3.first_name) ||' '|| INITCAP(p3.last_name)), ' | ') WITHIN GROUP (ORDER BY s.nurse_id)))||']' AS Medical_staff
  4  FROM servicerecord s, admission a, patient pt, person p, doctor d, employee e, person p2, nurse n, employee e2, person p3
  5  WHERE s.admission_id=a.admission_id
  6  AND a.patient_id=pt.patient_id
  7  AND pt.patient_id=p.person_id
  8  AND s.doctor_id=d.doctor_id
  9  AND d.doctor_id=e.employee_id
 10  AND e.employee_id=p2.person_id
 11  AND s.nurse_id=n.nurse_id(+)
 12  AND n.nurse_id=e2.employee_id(+)
 13  AND e2.employee_id=p3.person_id(+)
 14  AND SYSDATE - CAST(a.admission_date AS DATE) <= &days
 15  AND SYSDATE - CAST(a.admission_date AS DATE) > 0
 16  AND a.bed_id IS NOT NULL
 17  AND a.status != 'O'
 18  AND discharge_date IS NULL
 19  GROUP BY s.admission_id,pt.patient_id,p.first_name,p.last_name, s.admission_id;
Enter value for days: 5
old  14: AND SYSDATE - CAST(a.admission_date AS DATE) <= &days
new  14: AND SYSDATE - CAST(a.admission_date AS DATE) <= 5

PATIENT                                 ADMISS FIRST_SERVICE_TIME                          MEDICAL_STAFF
--------------------------------------- ------ ------------------------------------------- ------------------------------------------------------------------------------------------------------------------------------------
P00004, MerryYeung                      A00005 22-MAR-21 05.10.10.123000 PM                [P00021, Dr. Sasha Braus | P00022, Dr. Eren Yeager | P00023, Dr. Mikasa Ackerman ] && [P00017, Maggi Nairn | P00018, Genni Rhys | P00019, Berkie Da
P00005, PuspaSangkara                   A00006 22-MAR-21 07.10.10.123000 PM                [P00024, Dr. Erwin Smith ] && [P00018, Genni Rhys]
P00001, Jheng KhinYap                   A00009 27-MAR-21 10.23.17.000000 AM                [P00023, Dr. Mikasa Ackerman ] && [P00016, Kay Fedoronko]
P00002, HaryatiIzzati                   A00010 24-MAR-21 10.23.17.000000 AM                [P00023, Dr. Mikasa Ackerman | P00023, Dr. Mikasa Ackerman ] && [P00016, Kay Fedoronko | P00017, Maggi Nairn]
```

For all patient, the nearest n day(s):999999



```
SQL> SELECT (pt.patient_id||', '|| INITCAP(p.first_name) ||''|| INITCAP(p.last_name)) AS Patient, s.admission_id, MIN(start_time)||' ' AS First_Service_Time,
  2  '['||((LISTAGG((d.doctor_id||', Dr. '||INITCAP(p2.first_name) ||' '|| INITCAP(p2.last_name)), ' | ') WITHIN GROUP (ORDER BY d.doctor_id))||' ')||'] && ['||
  3  ((LISTAGG((s.nurse_id||', '||INITCAP(p3.first_name) ||' '|| INITCAP(p3.last_name)), ' | ') WITHIN GROUP (ORDER BY s.nurse_id)))||']' AS Medical_staff
  4  FROM servicerecord s, admission a, patient pt, person p, doctor d, employee e, person p2, nurse n, employee e2, person p3
  5  WHERE s.admission_id=a.admission_id
  6  AND a.patient_id=pt.patient_id
  7  AND pt.patient_id=p.person_id
  8  AND s.doctor_id=d.doctor_id
  9  AND d.doctor_id=e.employee_id
 10  AND e.employee_id=p2.person_id
 11  AND s.nurse_id=n.nurse_id(+)
 12  AND n.nurse_id=e2.employee_id(+)
 13  AND e2.employee_id=p3.person_id(+)
 14  AND SYSDATE - CAST(a.admission_date AS DATE) <= &days
 15  AND SYSDATE - CAST(a.admission_date AS DATE) > 0
 16  AND a.bed_id IS NOT NULL
 17  AND a.status != 'O'
 18  AND discharge_date IS NULL
 19  GROUP BY s.admission_id,pt.patient_id,p.first_name,p.last_name, s.admission_id;
Enter value for days: 999999
old  14: AND SYSDATE - CAST(a.admission_date AS DATE) <= &days
new  14: AND SYSDATE - CAST(a.admission_date AS DATE) <= 999999

PATIENT                                 ADMISS FIRST_SERVICE_TIME                          MEDICAL_STAFF
--------------------------------------- ------ ------------------------------------------- ------------------------------------------------------------------------------------------------------------------------------------
P00003, KevinOwens                      A00004 21-MAR-21 05.10.10.123000 PM                [P00022, Dr. Eren Yeager | P00022, Dr. Eren Yeager | P00023, Dr. Mikasa Ackerman | P00023, Dr. Mikasa Ackerman | P00023, Dr. Mikasa Ackerman | P000
P00004, MerryYeung                      A00005 22-MAR-21 05.10.10.123000 PM                [P00021, Dr. Sasha Braus | P00022, Dr. Eren Yeager | P00023, Dr. Mikasa Ackerman ] && [P00017, Maggi Nairn | P00018, Genni Rhys | P00019, Berkie Da
P00005, PuspaSangkara                   A00006 22-MAR-21 07.10.10.123000 PM                [P00024, Dr. Erwin Smith ] && [P00018, Genni Rhys]
P00001, Jheng KhinYap                   A00009 27-MAR-21 10.23.17.000000 AM                [P00023, Dr. Mikasa Ackerman ] && [P00016, Kay Fedoronko]
P00002, HaryatiIzzati                   A00010 24-MAR-21 10.23.17.000000 AM                [P00023, Dr. Mikasa Ackerman | P00023, Dr. Mikasa Ackerman ] && [P00016, Kay Fedoronko | P00017, Maggi Nairn]
P00032, RickeyLiles                     A00018 15-MAR-21 10.00.10.123000 AM                [P00024, Dr. Erwin Smith ] && [,  ]

6 rows selected.
```

Query 6

<u>Show all historical medicine equipment undertaking by a patient</u>

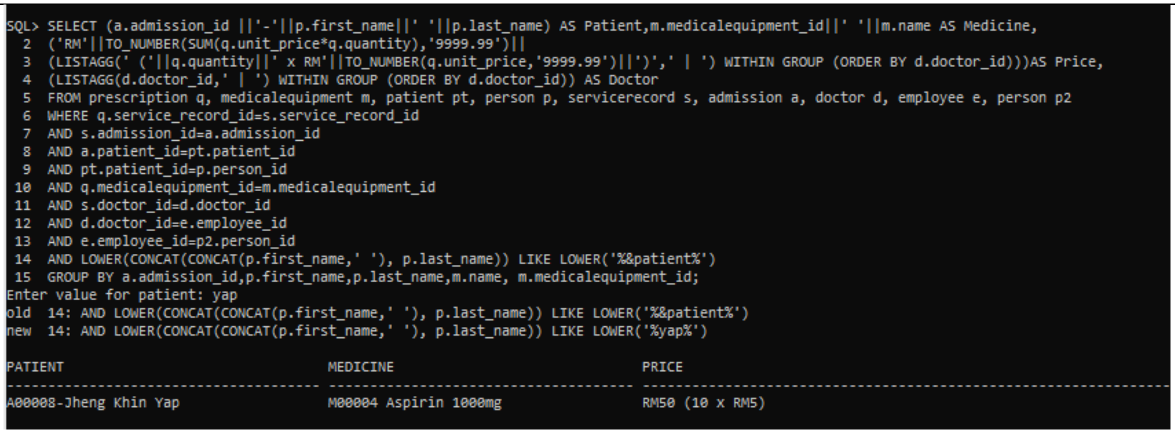
- A check list for nurse or staff to prepare medicine or medicine equipment at counter
- The query will prompt user to input full or patient name (not case sensitive) to query.
- This is helpful for nurse or doctor to check patient has taking what medicine by other doctor so that no need to give the same medicine again. A price also displays out if the patient discharge and this query can use to count the finalized medicine equipment with detail display each dosage given by different doctor in each service they provided (This can be achieved by using ‘WHERE(AND) discharge_date is NULL’.

<u>SQL command</u>

```
SELECT (a.admission_id ||'-'||p.first_name||' '||p.last_name) AS Patient,m.medicalequipment_id||' '||m.name AS Medicine,
('RM'||TO_NUMBER(SUM(q.unit_price*q.quantity),'9999.99')||
(LISTAGG(' ('||q.quantity||' x RM'||TO_NUMBER(q.unit_price,'9999.99')||')',' | ') WITHIN GROUP (ORDER BY
d.doctor_id)))AS Price,
(LISTAGG(d.doctor_id,' | ') WITHIN GROUP (ORDER BY d.doctor_id)) AS Doctor
FROM prescription q, medicalequipment m, patient pt, person p, servicerecord s, admission a, doctor d, employee e,
person p2
WHERE q.service_record_id=s.service_record_id
AND s.admission_id=a.admission_id
AND a.patient_id=pt.patient_id
AND pt.patient_id=p.person_id
AND q.medicalequipment_id=m.medicalequipment_id
AND s.doctor_id=d.doctor_id
AND d.doctor_id=e.employee_id
AND e.employee_id=p2.person_id
AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
GROUP BY a.admission_id,p.first_name,p.last_name,m.name, m.medicalequipment_id;
```

<u>Screenshot</u>

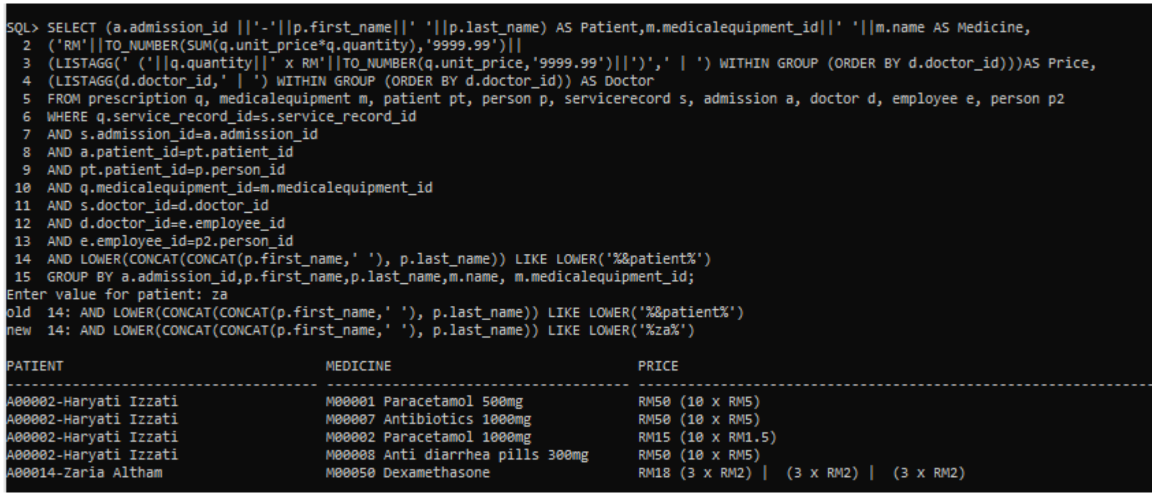
Using patient name: yap



```
SQL> SELECT (a.admission_id ||'-'||p.first_name||' '||p.last_name) AS Patient,m.medicalequipment_id||' '||m.name AS Medicine,
  2  ('RM'||TO_NUMBER(SUM(q.unit_price*q.quantity),'9999.99')||
  3  (LISTAGG(' ('||q.quantity||' x RM'||TO_NUMBER(q.unit_price,'9999.99')||')',' | ') WITHIN GROUP (ORDER BY d.doctor_id)))AS Price,
  4  (LISTAGG(d.doctor_id,' | ') WITHIN GROUP (ORDER BY d.doctor_id)) AS Doctor
  5  FROM prescription q, medicalequipment m, patient pt, person p, servicerecord s, admission a, doctor d, employee e, person p2
  6  WHERE q.service_record_id=s.service_record_id
  7  AND s.admission_id=a.admission_id
  8  AND a.patient_id=pt.patient_id
  9  AND pt.patient_id=p.person_id
 10  AND q.medicalequipment_id=m.medicalequipment_id
 11  AND s.doctor_id=d.doctor_id
 12  AND d.doctor_id=e.employee_id
 13  AND e.employee_id=p2.person_id
 14  AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
 15  GROUP BY a.admission_id,p.first_name,p.last_name,m.name, m.medicalequipment_id;
Enter value for patient: yap
old  14: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
new  14: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%yap%')

PATIENT                              MEDICINE                             PRICE
------------------------------------ ------------------------------------ --------------------------------------------------------------
A00008-Jheng Khin Yap                M00004 Aspirin 1000mg                RM50 (10 x RM5)
```

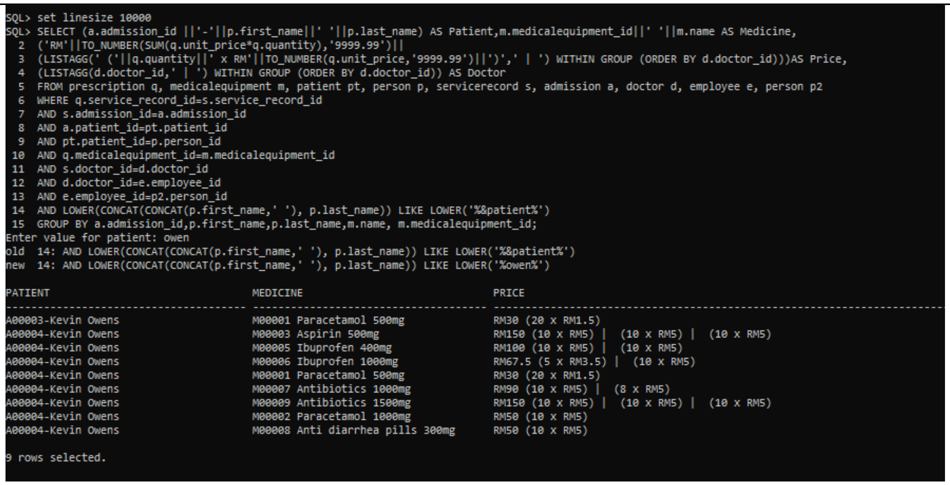
Using patient name: za
Two names consist of ‘za’ is shown



```
SQL> SELECT (a.admission_id ||'-'||p.first_name||' '||p.last_name) AS Patient,m.medicalequipment_id||' '||m.name AS Medicine,
  2  ('RM'||TO_NUMBER(SUM(q.unit_price*q.quantity),'9999.99')||
  3  (LISTAGG(' ('||q.quantity||' x RM'||TO_NUMBER(q.unit_price,'9999.99')||')',' | ') WITHIN GROUP (ORDER BY d.doctor_id)))AS Price,
  4  (LISTAGG(d.doctor_id,' | ') WITHIN GROUP (ORDER BY d.doctor_id)) AS Doctor
  5  FROM prescription q, medicalequipment m, patient pt, person p, servicerecord s, admission a, doctor d, employee e, person p2
  6  WHERE q.service_record_id=s.service_record_id
  7  AND s.admission_id=a.admission_id
  8  AND a.patient_id=pt.patient_id
  9  AND pt.patient_id=p.person_id
 10  AND q.medicalequipment_id=m.medicalequipment_id
 11  AND s.doctor_id=d.doctor_id
 12  AND d.doctor_id=e.employee_id
 13  AND e.employee_id=p2.person_id
 14  AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
 15  GROUP BY a.admission_id,p.first_name,p.last_name,m.name, m.medicalequipment_id;
Enter value for patient: za
old  14: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
new  14: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%za%')

PATIENT                              MEDICINE                             PRICE
------------------------------------ ------------------------------------ --------------------------------------------------------------
A00002-Haryati Izzati                M00001 Paracetamol 500mg             RM50 (10 x RM5)
A00002-Haryati Izzati                M00007 Antibiotics 1000mg            RM50 (10 x RM5)
A00002-Haryati Izzati                M00002 Paracetamol 1000mg            RM15 (10 x RM1.5)
A00002-Haryati Izzati                M00008 Anti diarrhea pills 300mg     RM50 (10 x RM5)
A00014-Zaria Altham                  M00050 Dexamethasone                 RM18 (3 x RM2) |  (3 x RM2) |  (3 x RM2)
```

Using patient name: owen
This can show that all medical history is prompted out regardless of admission for doctor reference purpose

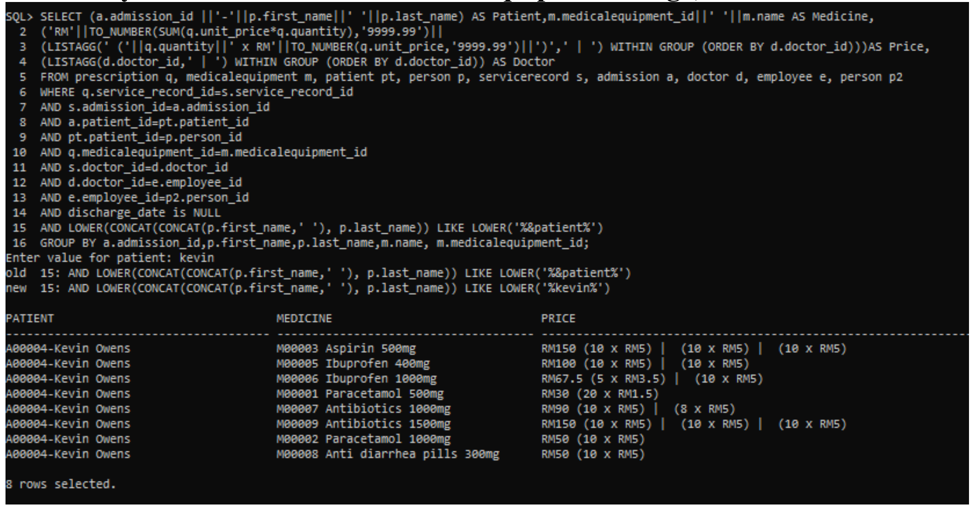


```
SQL> set linesize 10000
SQL> SELECT (a.admission_id ||'-'||p.first_name||' '||p.last_name) AS Patient,m.medicalequipment_id||' '||m.name AS Medicine,
  2  ('RM'||TO_NUMBER(SUM(q.unit_price*q.quantity),'9999.99')||
  3  (LISTAGG(' ('||q.quantity||' x RM'||TO_NUMBER(q.unit_price,'9999.99')||')',' | ') WITHIN GROUP (ORDER BY d.doctor_id)))AS Price,
  4  (LISTAGG(d.doctor_id,' | ') WITHIN GROUP (ORDER BY d.doctor_id)) AS Doctor
  5  FROM prescription q, medicalequipment m, patient pt, person p, servicerecord s, admission a, doctor d, employee e, person p2
  6  WHERE q.service_record_id=s.service_record_id
  7  AND s.admission_id=a.admission_id
  8  AND a.patient_id=pt.patient_id
  9  AND pt.patient_id=p.person_id
 10  AND q.medicalequipment_id=m.medicalequipment_id
 11  AND s.doctor_id=d.doctor_id
 12  AND d.doctor_id=e.employee_id
 13  AND e.employee_id=p2.person_id
 14  AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
 15  GROUP BY a.admission_id,p.first_name,p.last_name,m.name, m.medicalequipment_id;
Enter value for patient: owen
old  14: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
new  14: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%owen%')

PATIENT                              MEDICINE                             PRICE
------------------------------------ ------------------------------------ ----------------------------------------------------------------------------
A00003-Kevin Owens                   M00001 Paracetamol 500mg             RM30 (20 x RM1.5)
A00004-Kevin Owens                   M00003 Aspirin 500mg                 RM150 (10 x RM5) |  (10 x RM5) |  (10 x RM5)
A00004-Kevin Owens                   M00005 Ibuprofen 400mg               RM100 (10 x RM5) |  (10 x RM5)
A00004-Kevin Owens                   M00006 Ibuprofen 1000mg              RM67.5 (5 x RM3.5) |  (10 x RM5)
A00004-Kevin Owens                   M00001 Paracetamol 500mg             RM30 (20 x RM1.5)
A00004-Kevin Owens                   M00007 Antibiotics 1000mg            RM90 (10 x RM5) |  (8 x RM5)
A00004-Kevin Owens                   M00009 Antibiotics 1500mg            RM150 (10 x RM5) |  (10 x RM5) |  (10 x RM5)
A00004-Kevin Owens                   M00002 Paracetamol 1000mg            RM50 (10 x RM5)
A00004-Kevin Owens                   M00008 Anti diarrhea pills 300mg     RM50 (10 x RM5)

9 rows selected.
```

Using patient name: owen
Show only current admission medical equipment usage, as added ‘AND discharge_date is NULL’.



```
SQL> SELECT (a.admission_id ||'-'||p.first_name||' '||p.last_name) AS Patient,m.medicalequipment_id||' '||m.name AS Medicine,
  2  ('RM'||TO_NUMBER(SUM(q.unit_price*q.quantity),'9999.99')||
  3  (LISTAGG(' ('||q.quantity||' x RM'||TO_NUMBER(q.unit_price,'9999.99')||')',' | ') WITHIN GROUP (ORDER BY d.doctor_id)))AS Price,
  4  (LISTAGG(d.doctor_id,' | ') WITHIN GROUP (ORDER BY d.doctor_id)) AS Doctor
  5  FROM prescription q, medicalequipment m, patient pt, person p, servicerecord s, admission a, doctor d, employee e, person p2
  6  WHERE q.service_record_id=s.service_record_id
  7  AND s.admission_id=a.admission_id
  8  AND a.patient_id=pt.patient_id
  9  AND pt.patient_id=p.person_id
 10  AND q.medicalequipment_id=m.medicalequipment_id
 11  AND s.doctor_id=d.doctor_id
 12  AND d.doctor_id=e.employee_id
 13  AND e.employee_id=p2.person_id
 14  AND discharge_date is NULL
 15  AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
 16  GROUP BY a.admission_id,p.first_name,p.last_name,m.name, m.medicalequipment_id;
Enter value for patient: kevin
old  15: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%&patient%')
new  15: AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER('%kevin%')

PATIENT                              MEDICINE                             PRICE
------------------------------------ ------------------------------------ ----------------------------------------------------------------------------
A00004-Kevin Owens                   M00003 Aspirin 500mg                 RM150 (10 x RM5) |  (10 x RM5) |  (10 x RM5)
A00004-Kevin Owens                   M00005 Ibuprofen 400mg               RM100 (10 x RM5) |  (10 x RM5)
A00004-Kevin Owens                   M00006 Ibuprofen 1000mg              RM67.5 (5 x RM3.5) |  (10 x RM5)
A00004-Kevin Owens                   M00001 Paracetamol 500mg             RM30 (20 x RM1.5)
A00004-Kevin Owens                   M00007 Antibiotics 1000mg            RM90 (10 x RM5) |  (8 x RM5)
A00004-Kevin Owens                   M00009 Antibiotics 1500mg            RM150 (10 x RM5) |  (10 x RM5) |  (10 x RM5)
A00004-Kevin Owens                   M00002 Paracetamol 1000mg            RM50 (10 x RM5)
A00004-Kevin Owens                   M00008 Anti diarrhea pills 300mg     RM50 (10 x RM5)

8 rows selected.
```

Query 7	<u>**Show operation room schedule by entering room name**</u> • A check list for important room such as Operation Theatre. It will order by the timing start from earlier	

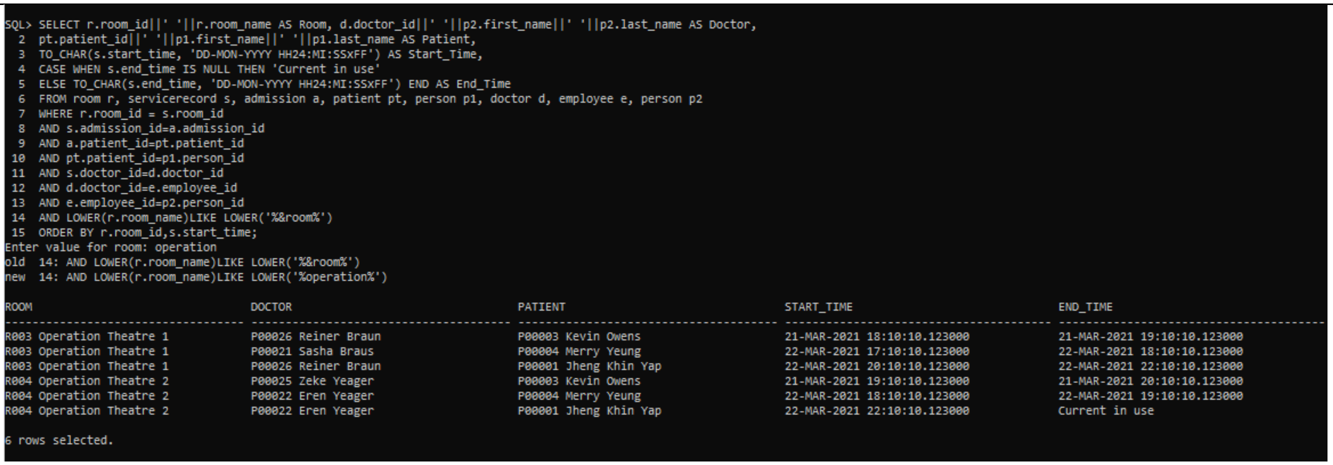
- This query also can show all related room in once by keyword. For instance, ‘operation’ can list all operation room1 and room 2 while it will arrange by a group of same room and follow by the timing.
- This check query can check the arrangement of operation theatre and show the related doctor and patient. It can prevent the collision of two patient using the same room.
- The query also useful to check who are current in use of the room, so that receptionist can tell the patient family member the location of patient in which operation room.

SQL command

```
SELECT r.room_id||' '||r.room_name AS Room, d.doctor_id||' '||p2.first_name||' '||p2.last_name AS Doctor,
pt.patient_id||' '||p1.first_name||' '||p1.last_name AS Patient,
TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI:SSxFF') AS Start_Time,
CASE WHEN s.end_time IS NULL THEN 'Current in use'
ELSE TO_CHAR(s.end_time, 'DD-MON-YYYY HH24:MI:SSxFF') END AS End_Time
FROM room r, servicerecord s, admission a, patient pt, person p1, doctor d, employee e, person p2
WHERE r.room_id = s.room_id
AND s.admission_id=a.admission_id
AND a.patient_id=pt.patient_id
AND pt.patient_id=p1.person_id
AND s.doctor_id=d.doctor_id
AND d.doctor_id=e.employee_id
AND e.employee_id=p2.person_id
AND LOWER(r.room_name)LIKE LOWER('%&room%')
ORDER BY r.room_id,s.start_time;
```

Screenshot

Using room name: Operation

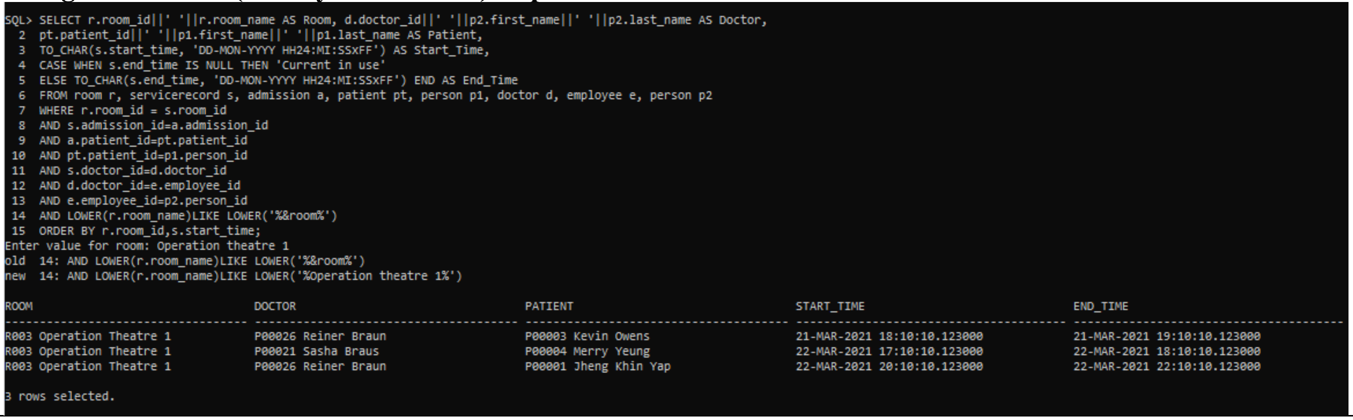


```
SQL> SELECT r.room_id||' '||r.room_name AS Room, d.doctor_id||' '||p2.first_name||' '||p2.last_name AS Doctor,
  2  pt.patient_id||' '||p1.first_name||' '||p1.last_name AS Patient,
  3  TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI:SSxFF') AS Start_Time,
  4  CASE WHEN s.end_time IS NULL THEN 'Current in use'
  5  ELSE TO_CHAR(s.end_time, 'DD-MON-YYYY HH24:MI:SSxFF') END AS End_Time
  6  FROM room r, servicerecord s, admission a, patient pt, person p1, doctor d, employee e, person p2
  7  WHERE r.room_id = s.room_id
  8  AND s.admission_id=a.admission_id
  9  AND a.patient_id=pt.patient_id
 10  AND pt.patient_id=p1.person_id
 11  AND s.doctor_id=d.doctor_id
 12  AND d.doctor_id=e.employee_id
 13  AND e.employee_id=p2.person_id
 14  AND LOWER(r.room_name)LIKE LOWER('%&room%')
 15  ORDER BY r.room_id,s.start_time;
Enter value for room: operation
old  14: AND LOWER(r.room_name)LIKE LOWER('%&room%')
new  14: AND LOWER(r.room_name)LIKE LOWER('%operation%')

ROOM                                DOCTOR                                PATIENT                               START_TIME                            END_TIME
----------------------------------- ------------------------------------- ------------------------------------- ------------------------------------- -------------------------------------
R003 Operation Theatre 1            P00026 Reiner Braun                   P00003 Kevin Owens                    21-MAR-2021 18:10:10.123000           21-MAR-2021 19:10:10.123000
R003 Operation Theatre 1            P00021 Sasha Braus                    P00004 Merry Yeung                    22-MAR-2021 17:10:10.123000           22-MAR-2021 18:10:10.123000
R003 Operation Theatre 1            P00026 Reiner Braun                   P00001 Jheng Khin Yap                 22-MAR-2021 20:10:10.123000           22-MAR-2021 22:10:10.123000
R004 Operation Theatre 2            P00025 Zeke Yeager                    P00003 Kevin Owens                    21-MAR-2021 19:10:10.123000           21-MAR-2021 20:10:10.123000
R004 Operation Theatre 2            P00022 Eren Yeager                    P00004 Merry Yeung                    22-MAR-2021 18:10:10.123000           22-MAR-2021 19:10:10.123000
R004 Operation Theatre 2            P00022 Eren Yeager                    P00001 Jheng Khin Yap                 22-MAR-2021 22:10:10.123000           Current in use

6 rows selected.
```

Using room name (exactly room name): Operation theatre 1



```
SQL> SELECT r.room_id||' '||r.room_name AS Room, d.doctor_id||' '||p2.first_name||' '||p2.last_name AS Doctor,
  2  pt.patient_id||' '||p1.first_name||' '||p1.last_name AS Patient,
  3  TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI:SSxFF') AS Start_Time,
  4  CASE WHEN s.end_time IS NULL THEN 'Current in use'
  5  ELSE TO_CHAR(s.end_time, 'DD-MON-YYYY HH24:MI:SSxFF') END AS End_Time
  6  FROM room r, servicerecord s, admission a, patient pt, person p1, doctor d, employee e, person p2
  7  WHERE r.room_id = s.room_id
  8  AND s.admission_id=a.admission_id
  9  AND a.patient_id=pt.patient_id
 10  AND pt.patient_id=p1.person_id
 11  AND s.doctor_id=d.doctor_id
 12  AND d.doctor_id=e.employee_id
 13  AND e.employee_id=p2.person_id
 14  AND LOWER(r.room_name)LIKE LOWER('%&room%')
 15  ORDER BY r.room_id,s.start_time;
Enter value for room: Operation theatre 1
old  14: AND LOWER(r.room_name)LIKE LOWER('%&room%')
new  14: AND LOWER(r.room_name)LIKE LOWER('%Operation theatre 1%')

ROOM                                DOCTOR                                PATIENT                               START_TIME                            END_TIME
----------------------------------- ------------------------------------- ------------------------------------- ------------------------------------- -------------------------------------
R003 Operation Theatre 1            P00026 Reiner Braun                   P00003 Kevin Owens                    21-MAR-2021 18:10:10.123000           21-MAR-2021 19:10:10.123000
R003 Operation Theatre 1            P00021 Sasha Braus                    P00004 Merry Yeung                    22-MAR-2021 17:10:10.123000           22-MAR-2021 18:10:10.123000
R003 Operation Theatre 1            P00026 Reiner Braun                   P00001 Jheng Khin Yap                 22-MAR-2021 20:10:10.123000           22-MAR-2021 22:10:10.123000

3 rows selected.
```

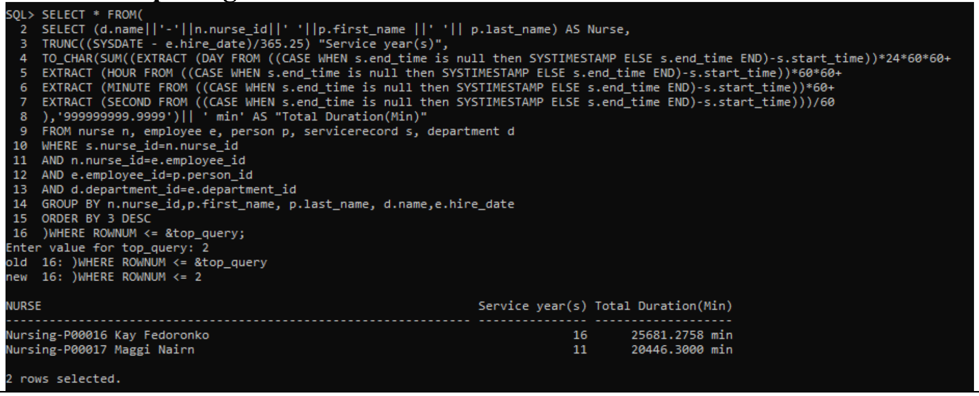
Query 8	**<u>Show nurse with highest service duration</u>** • This query can show out the highest nurse service minutes perform in his or her work. • This query can arrange the "nurse of the year". This can give motivation to nurse by reward. • This query will prompt to insert select among the top n highest service duration. • This query using nested select, after a list of duration is counted out and being order, the filter row query is ran to get the highest service duration.	

SQL command

```
SELECT * FROM(
SELECT (d.name||'-'||n.nurse_id||' '||p.first_name ||' '|| p.last_name) AS Nurse,
TRUNC((SYSDATE - e.hire_date)/365.25) "Service year(s)",
TO_CHAR(SUM((EXTRACT (DAY FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE
s.end_time END)-s.start_time))*24*60*60+
EXTRACT (HOUR FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-
s.start_time))*60*60+
EXTRACT (MINUTE FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-
s.start_time))*60+
EXTRACT (SECOND FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-
s.start_time)))/60
),'999999999.9999')|| ' min' AS "Total Duration(Min)"
FROM nurse n, employee e, person p, servicerecord s, department d
WHERE s.nurse_id=n.nurse_id
AND n.nurse_id=e.employee_id
AND e.employee_id=p.person_id
AND d.department_id=e.department_id
GROUP BY n.nurse_id,p.first_name, p.last_name, d.name,e.hire_date
ORDER BY 3 DESC
)WHERE ROWNUM <= &top_query;
```

Screenshot

Show the top 2 highest service duration result

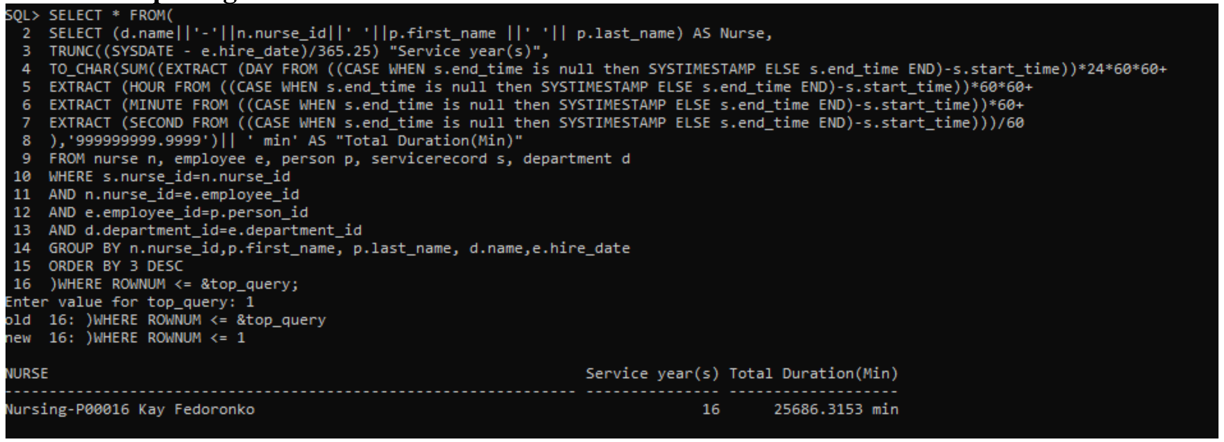


```
SQL> SELECT * FROM(
  2  SELECT (d.name||'-'||n.nurse_id||' '||p.first_name ||' '|| p.last_name) AS Nurse,
  3  TRUNC((SYSDATE - e.hire_date)/365.25) "Service year(s)",
  4  TO_CHAR(SUM((EXTRACT (DAY FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*24*60*60+
  5  EXTRACT (HOUR FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*60*60+
  6  EXTRACT (MINUTE FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*60+
  7  EXTRACT (SECOND FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time)))/60
  8  ),'999999999.9999')|| ' min' AS "Total Duration(Min)"
  9  FROM nurse n, employee e, person p, servicerecord s, department d
 10  WHERE s.nurse_id=n.nurse_id
 11  AND n.nurse_id=e.employee_id
 12  AND e.employee_id=p.person_id
 13  AND d.department_id=e.department_id
 14  GROUP BY n.nurse_id,p.first_name, p.last_name, d.name,e.hire_date
 15  ORDER BY 3 DESC
 16  )WHERE ROWNUM <= &top_query;
Enter value for top_query: 2
old  16: )WHERE ROWNUM <= &top_query
new  16: )WHERE ROWNUM <= 2

NURSE                                                            Service year(s) Total Duration(Min)
---------------------------------------------------------------- --------------- -------------------
Nursing-P00016 Kay Fedoronko                                                  16      25681.2758 min
Nursing-P00017 Maggi Nairn                                                    11      20446.3000 min

2 rows selected.
```

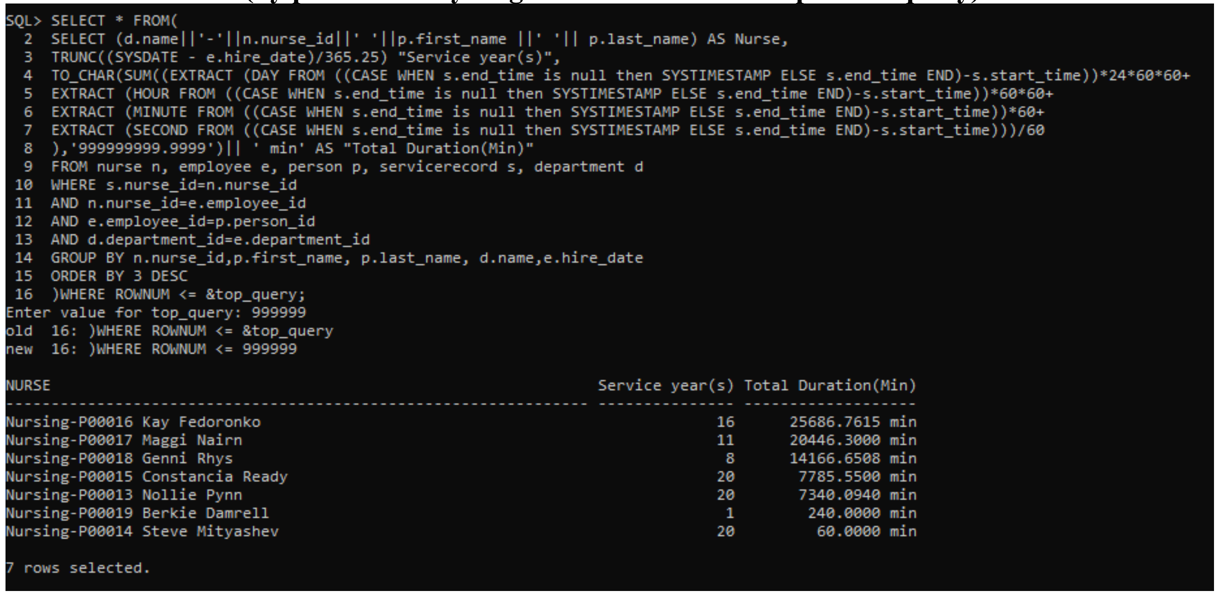
Show the top 1 highest service duration result



```
SQL> SELECT * FROM(
  2  SELECT (d.name||'-'||n.nurse_id||' '||p.first_name ||' '|| p.last_name) AS Nurse,
  3  TRUNC((SYSDATE - e.hire_date)/365.25) "Service year(s)",
  4  TO_CHAR(SUM((EXTRACT (DAY FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*24*60*60+
  5  EXTRACT (HOUR FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*60*60+
  6  EXTRACT (MINUTE FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*60+
  7  EXTRACT (SECOND FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time)))/60
  8  ),'999999999.9999')|| ' min' AS "Total Duration(Min)"
  9  FROM nurse n, employee e, person p, servicerecord s, department d
 10  WHERE s.nurse_id=n.nurse_id
 11  AND n.nurse_id=e.employee_id
 12  AND e.employee_id=p.person_id
 13  AND d.department_id=e.department_id
 14  GROUP BY n.nurse_id,p.first_name, p.last_name, d.name,e.hire_date
 15  ORDER BY 3 DESC
 16  )WHERE ROWNUM <= &top_query;
Enter value for top_query: 1
old  16: )WHERE ROWNUM <= &top_query
new  16: )WHERE ROWNUM <= 1

NURSE                                                            Service year(s) Total Duration(Min)
---------------------------------------------------------------- --------------- ------------------
Nursing-P00016 Kay Fedoronko                                                  16      25686.3153 min
```

Show all the nurse (by put extremely large number for select top result query)



```
SQL> SELECT * FROM(
  2  SELECT (d.name||'-'||n.nurse_id||' '||p.first_name ||' '|| p.last_name) AS Nurse,
  3  TRUNC((SYSDATE - e.hire_date)/365.25) "Service year(s)",
  4  TO_CHAR(SUM((EXTRACT (DAY FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*24*60*60+
  5  EXTRACT (HOUR FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*60*60+
  6  EXTRACT (MINUTE FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time))*60+
  7  EXTRACT (SECOND FROM ((CASE WHEN s.end_time is null then SYSTIMESTAMP ELSE s.end_time END)-s.start_time)))/60
  8  ),'999999999.9999')|| ' min' AS "Total Duration(Min)"
  9  FROM nurse n, employee e, person p, servicerecord s, department d
 10  WHERE s.nurse_id=n.nurse_id
 11  AND n.nurse_id=e.employee_id
 12  AND e.employee_id=p.person_id
 13  AND d.department_id=e.department_id
 14  GROUP BY n.nurse_id,p.first_name, p.last_name, d.name,e.hire_date
 15  ORDER BY 3 DESC
 16  )WHERE ROWNUM <= &top_query;
Enter value for top_query: 999999
old  16: )WHERE ROWNUM <= &top_query
new  16: )WHERE ROWNUM <= 999999

NURSE                                                            Service year(s) Total Duration(Min)
---------------------------------------------------------------- --------------- ------------------
Nursing-P00016 Kay Fedoronko                                                  16      25686.7615 min
Nursing-P00017 Maggi Nairn                                                    11      20446.3000 min
Nursing-P00018 Genni Rhys                                                      8      14166.6508 min
Nursing-P00015 Constancia Ready                                               20       7785.5500 min
Nursing-P00013 Nollie Pynn                                                    20       7340.0940 min
Nursing-P00019 Berkie Damrell                                                  1        240.0000 min
Nursing-P00014 Steve Mityashev                                                20         60.0000 min

7 rows selected.
```

Query 9

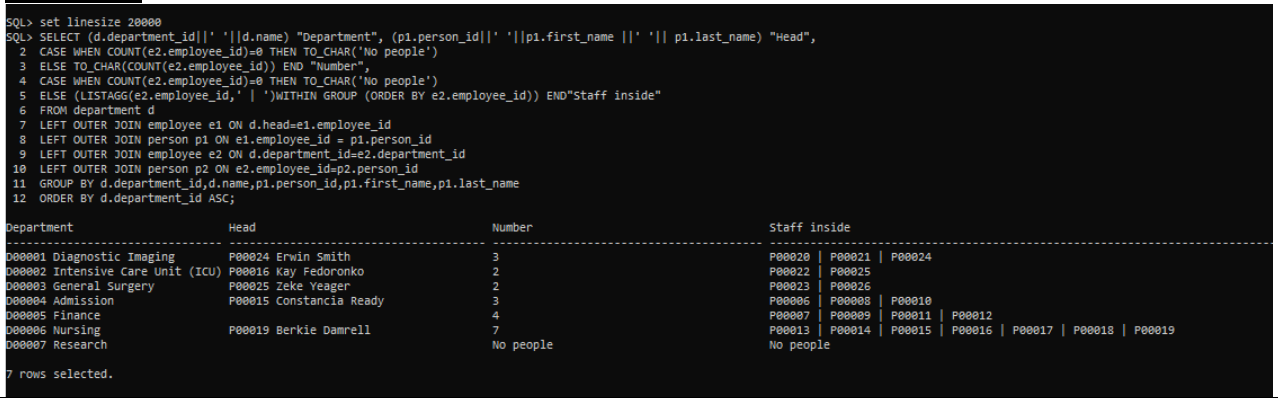
<u>Show the list of staff number in each department</u>

- This query can list out the total number of staff in each department
- This query is important for Human resources department staff to do analysis and recruit new doctor, nurse or employee if needed.
- This query can be added some condition to more filter sort out respective department data.
- This query using case to show 'No people' instead to show NULL value on the output

<u>SQL command</u>

```
SELECT (d.department_id||' '||d.name) "Department", (p1.person_id||' '||p1.first_name ||' '|| p1.last_name) "Head",
CASE WHEN COUNT(e2.employee_id)=0 THEN TO_CHAR('No people')
ELSE TO_CHAR(COUNT(e2.employee_id)) END "Number",
CASE WHEN COUNT(e2.employee_id)=0 THEN TO_CHAR('No people')
ELSE (LISTAGG(e2.employee_id,' | ')WITHIN GROUP (ORDER BY e2.employee_id)) END"Staff inside"
FROM department d
LEFT OUTER JOIN employee e1 ON d.head=e1.employee_id
LEFT OUTER JOIN person p1 ON e1.employee_id = p1.person_id
LEFT OUTER JOIN employee e2 ON d.department_id=e2.department_id
LEFT OUTER JOIN person p2 ON e2.employee_id=p2.person_id
GROUP BY d.department_id,d.name,p1.person_id,p1.first_name,p1.last_name
ORDER BY d.department_id ASC;
```

<u>Screenshot</u>



```
SQL> set linesize 20000
SQL> SELECT (d.department_id||' '||d.name) "Department", (p1.person_id||' '||p1.first_name ||' '|| p1.last_name) "Head",
     CASE WHEN COUNT(e2.employee_id)=0 THEN TO_CHAR('No people')
     ELSE TO_CHAR(COUNT(e2.employee_id)) END "Number",
     CASE WHEN COUNT(e2.employee_id)=0 THEN TO_CHAR('No people')
     ELSE (LISTAGG(e2.employee_id,' | ')WITHIN GROUP (ORDER BY e2.employee_id)) END"Staff inside"
     FROM department d
     LEFT OUTER JOIN employee e1 ON d.head=e1.employee_id
     LEFT OUTER JOIN person p1 ON e1.employee_id = p1.person_id
     LEFT OUTER JOIN employee e2 ON d.department_id=e2.department_id
     LEFT OUTER JOIN person p2 ON e2.employee_id=p2.person_id
     GROUP BY d.department_id,d.name,p1.person_id,p1.first_name,p1.last_name
     ORDER BY d.department_id ASC;

Department                       Head                                  Number                                  Staff inside
-------------------------------- ------------------------------------- --------------------------------------- ------------------------------------------------------------
D00001 Diagnostic Imaging        P00024 Erwin Smith                    3                                       P00020 | P00021 | P00024
D00002 Intensive Care Unit (ICU) P00016 Kay Fedoronko                  2                                       P00022 | P00025
D00003 General Surgery           P00025 Zeke Yeager                    2                                       P00023 | P00026
D00004 Admission                 P00015 Constancia Ready               3                                       P00006 | P00008 | P00010
D00005 Finance                                                         4                                       P00007 | P00009 | P00011 | P00012
D00006 Nursing                   P00019 Berkie Damrell                 7                                       P00013 | P00014 | P00015 | P00016 | P00017 | P00018 | P00019
D00007 Research                                                        No people                               No people

7 rows selected.
```

Query 10

<u>Show the list of patients that a nurse served on the day</u>

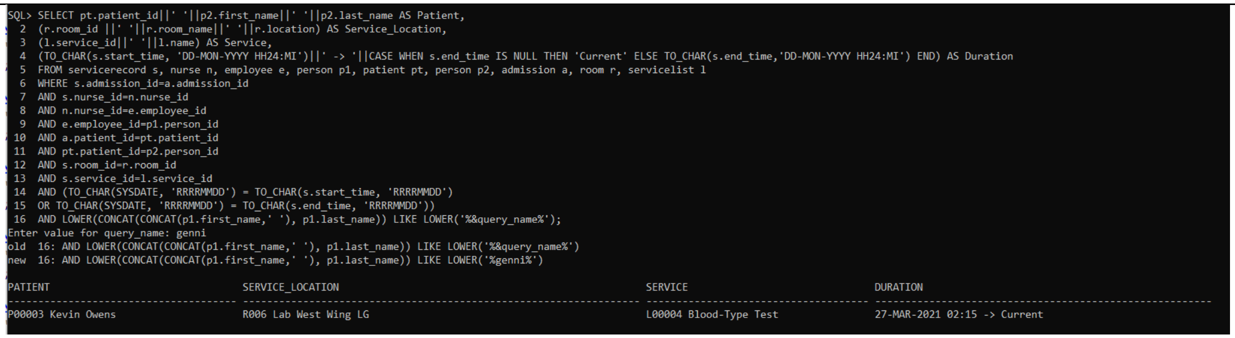
- This query can list out service that a nurse involved. The detail such as patient name, service location and service type is shown by this query
- In this pandemic situation, nurse also can trace herself if he or she has close contacted to covid-19 patient before.
- This query can be altered by changing SYSDATE to SYSDATE+(n) to search respective date service.
- This query can help nurse to make patrol or checking to his or her patient to make sure the patient situation is normal
- This query is useful for nurse to do his or her summary report of the day.

<u>SQL command</u>

```
SELECT pt.patient_id||' '||p2.first_name||' '||p2.last_name AS Patient,
(r.room_id ||' '||r.room_name||' '||r.location) AS Service_Location,
(l.service_id||' '||l.name) AS Service,
(TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI')||' -> '||CASE WHEN s.end_time IS NULL THEN 'Current'
ELSE TO_CHAR(s.end_time,'DD-MON-YYYY HH24:MI') END) AS Duration
FROM servicerecord s, nurse n, employee e, person p1, patient pt, person p2, admission a, room r, servicelist l
WHERE s.admission_id=a.admission_id
AND s.nurse_id=n.nurse_id
AND n.nurse_id=e.employee_id
AND e.employee_id=p1.person_id
AND a.patient_id=pt.patient_id
AND pt.patient_id=p2.person_id
AND s.room_id=r.room_id
AND s.service_id=l.service_id
AND (TO_CHAR(SYSDATE, 'RRRRMMDD') = TO_CHAR(s.start_time, 'RRRRMMDD')
OR TO_CHAR(SYSDATE, 'RRRRMMDD') = TO_CHAR(s.end_time, 'RRRRMMDD'))
AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%&query_name%');
```

<u>Screenshot</u>

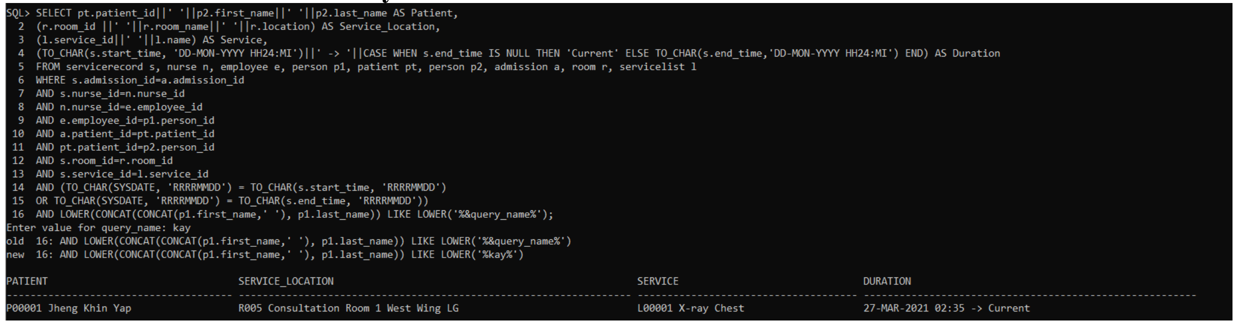
Search for a nurse's name: genni



```
SQL> SELECT pt.patient_id||' '||p2.first_name||' '||p2.last_name AS Patient,
  2  (r.room_id ||' '||r.room_name||' '||r.location) AS Service_Location,
  3  (l.service_id||' '||l.name) AS Service,
  4  (TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI')||' -> '||CASE WHEN s.end_time IS NULL THEN 'Current' ELSE TO_CHAR(s.end_time,'DD-MON-YYYY HH24:MI') END) AS Duration
  5  FROM servicerecord s, nurse n, employee e, person p1, patient pt, person p2, admission a, room r, servicelist l
  6  WHERE s.admission_id=a.admission_id
  7  AND s.nurse_id=n.nurse_id
  8  AND n.nurse_id=e.employee_id
  9  AND e.employee_id=p1.person_id
 10  AND a.patient_id=pt.patient_id
 11  AND pt.patient_id=p2.person_id
 12  AND s.room_id=r.room_id
 13  AND s.service_id=l.service_id
 14  AND (TO_CHAR(SYSDATE, 'RRRRMMDD') = TO_CHAR(s.start_time, 'RRRRMMDD')
 15  OR TO_CHAR(SYSDATE, 'RRRRMMDD') = TO_CHAR(s.end_time, 'RRRRMMDD'))
 16  AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%&query_name%');
Enter value for query_name: genni
old  16: AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%&query_name%')
new  16: AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%genni%')

PATIENT                                SERVICE_LOCATION                                                   SERVICE                              DURATION
-------------------------------------- ------------------------------------------------------------------ ------------------------------------ ------------------------------------------------
P00003 Kevin Owens                     R006 Lab West Wing LG                                              L00004 Blood-Type Test               27-MAR-2021 02:15 -> Current
```

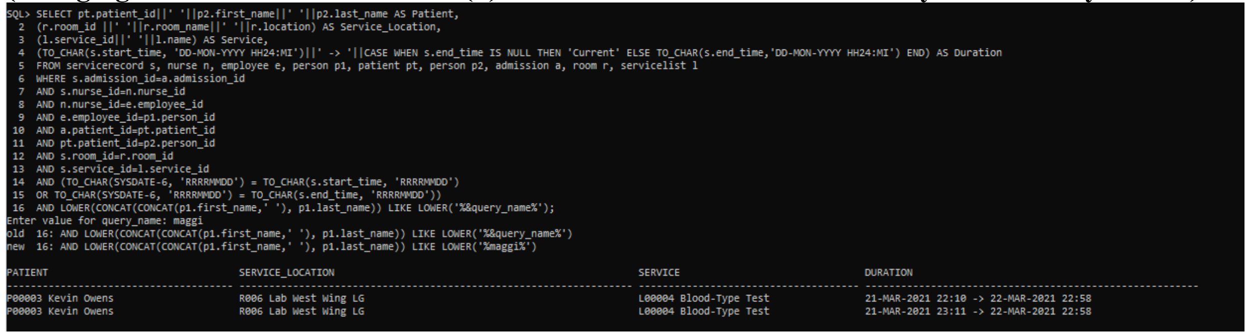
Search for a nurse's name: kay



```
SQL> SELECT pt.patient_id||' '||p2.first_name||' '||p2.last_name AS Patient,
  2  (r.room_id ||' '||r.room_name||' '||r.location) AS Service_Location,
  3  (l.service_id||' '||l.name) AS Service,
  4  (TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI')||' -> '||CASE WHEN s.end_time IS NULL THEN 'Current' ELSE TO_CHAR(s.end_time,'DD-MON-YYYY HH24:MI') END) AS Duration
  5  FROM servicerecord s, nurse n, employee e, person p1, patient pt, person p2, admission a, room r, servicelist l
  6  WHERE s.admission_id=a.admission_id
  7  AND s.nurse_id=n.nurse_id
  8  AND n.nurse_id=e.employee_id
  9  AND e.employee_id=p1.person_id
 10  AND a.patient_id=pt.patient_id
 11  AND pt.patient_id=p2.person_id
 12  AND s.room_id=r.room_id
 13  AND s.service_id=l.service_id
 14  AND (TO_CHAR(SYSDATE, 'RRRRMMDD') = TO_CHAR(s.start_time, 'RRRRMMDD')
 15  OR TO_CHAR(SYSDATE, 'RRRRMMDD') = TO_CHAR(s.end_time, 'RRRRMMDD'))
 16  AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%&query_name%');
Enter value for query_name: kay
old  16: AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%&query_name%')
new  16: AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%kay%')

PATIENT                                SERVICE_LOCATION                                                   SERVICE                              DURATION
-------------------------------------- ------------------------------------------------------------------ ------------------------------------ ------------------------------------------------
P00001 Jheng Khin Yap                  R005 Consultation Room 1 West Wing LG                              L00001 X-ray Chest                   27-MAR-2021 02:35 -> Current
```

Search for a nurse's name: maggi
(changing SYSDATE to SYSDATE+(n): SYSDATE-6 to search the 6th day before today record)



```
SQL> SELECT pt.patient_id||' '||p2.first_name||' '||p2.last_name AS Patient,
  2  (r.room_id ||' '||r.room_name||' '||r.location) AS Service_Location,
  3  (l.service_id||' '||l.name) AS Service,
  4  (TO_CHAR(s.start_time, 'DD-MON-YYYY HH24:MI')||' -> '||CASE WHEN s.end_time IS NULL THEN 'Current' ELSE TO_CHAR(s.end_time,'DD-MON-YYYY HH24:MI') END) AS Duration
  5  FROM servicerecord s, nurse n, employee e, person p1, patient pt, person p2, admission a, room r, servicelist l
  6  WHERE s.admission_id=a.admission_id
  7  AND s.nurse_id=n.nurse_id
  8  AND n.nurse_id=e.employee_id
  9  AND e.employee_id=p1.person_id
 10  AND a.patient_id=pt.patient_id
 11  AND pt.patient_id=p2.person_id
 12  AND s.room_id=r.room_id
 13  AND s.service_id=l.service_id
 14  AND (TO_CHAR(SYSDATE-6, 'RRRRMMDD') = TO_CHAR(s.start_time, 'RRRRMMDD')
 15  OR TO_CHAR(SYSDATE-6, 'RRRRMMDD') = TO_CHAR(s.end_time, 'RRRRMMDD'))
 16  AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%&query_name%');
Enter value for query_name: maggi
old  16: AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%&query_name%')
new  16: AND LOWER(CONCAT(CONCAT(p1.first_name,' '), p1.last_name)) LIKE LOWER('%maggi%')

PATIENT                                SERVICE_LOCATION                                                   SERVICE                              DURATION
-------------------------------------- ------------------------------------------------------------------ ------------------------------------ ------------------------------------------------
P00003 Kevin Owens                     R006 Lab West Wing LG                                              L00004 Blood-Type Test               21-MAR-2021 22:10 -> 22-MAR-2021 22:58
P00003 Kevin Owens                     R006 Lab West Wing LG                                              L00004 Blood-Type Test               21-MAR-2021 23:11 -> 22-MAR-2021 22:58
```

Stored Procedure (10 marks)		

SP1	**Check patient current situation. (Patient's location and what service undergo)**
	• This stored procedure able to check the patient situation including current room, current bed and current service and location.
	• When patient's family or friends come as visitor, nurse may tell them where to go to find the patient. In the meanwhile, nurse can directly tell the visitor they might need to wait as the patient is undergo some service. This is important to provide a good experience for visitor.
	• When hospital staff want to find where the patient located to do some follow up checking, this procedure can speedy and direct show where they can find the patient.
	
	Stored procedure
	`--Stored procedure`
	`CREATE OR REPLACE PROCEDURE patient_current_situation`
	`(`
	`  patient_name IN varchar2`
	`)`
	`IS`
	`    Admission Varchar2(6);`
	`    Bed varchar2(20);`
	`    Room varchar2(100);`
	`    Patient varchar2(100);`
	`    Nurse varchar2(100);`
	`    Doctor varchar2(100);`
	`    Service varchar2(100);`
	`    patient_n varchar2(200);`
	`    checkexist number(1);`
	`BEGIN`
	`    SELECT COUNT(*) INTO checkexist`
	`    FROM admission a, patient pt, person p`
	`    WHERE a.patient_id = pt.patient_id`
	`    AND pt.patient_id = p.person_id`
	`    AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER(patient_name)`	

```
    AND a.discharge_date IS NULL
    AND status != 'O';

    IF (checkexist=1) THEN
        SELECT a.admission_id, b.bed_id, (r.room_id ||' '||r.room_name||' '||r.location), (pt.patient_id||', '||p.first_name||'
'||p.last_name) INTO Admission, Bed, Room, Patient
        FROM patient pt, admission a, person p, bed b, room r
        WHERE a.patient_id=pt.patient_id AND pt.patient_id=p.person_id
        AND b.room_id=r.room_id
        AND a.bed_id=b.bed_id
        AND LOWER(CONCAT(CONCAT(p.first_name,' '), p.last_name)) LIKE LOWER(patient_name)
        AND a.discharge_date IS NULL;

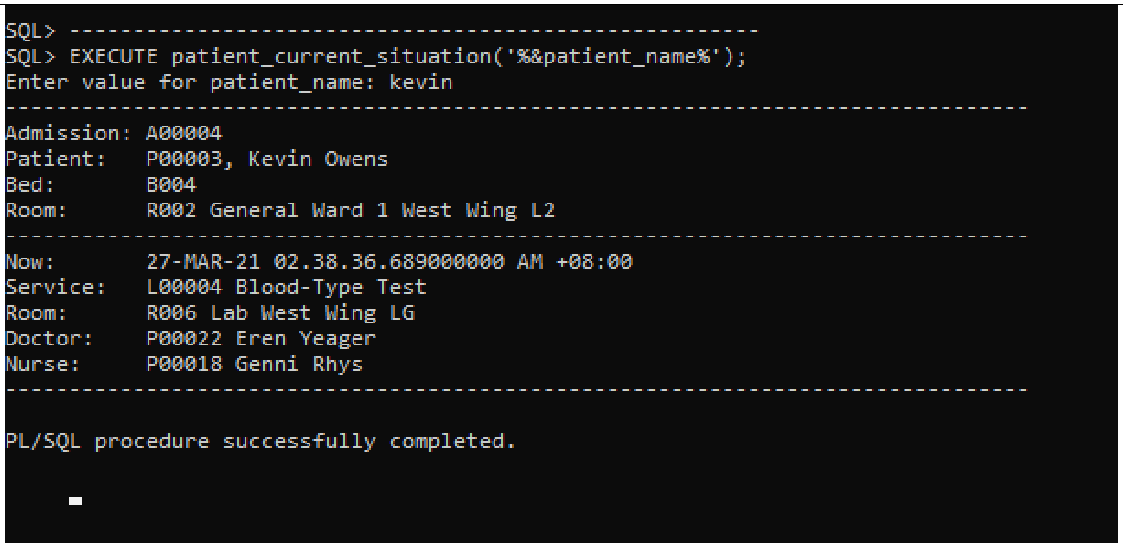
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('Admission: ' || Admission );
        DBMS_OUTPUT.PUT_LINE('Patient:   ' || Patient );
        DBMS_OUTPUT.PUT_LINE('Bed:       ' || Bed );
        DBMS_OUTPUT.PUT_LINE('Room:      ' || Room );

    SELECT COUNT(*) INTO checkexist FROM servicerecord sr WHERE sr.admission_id = Admission AND
end_time is NULL;

    IF (checkexist=1) THEN
        SELECT (r.room_id ||' '||r.room_name||' '||r.location), (s.nurse_id||' '||pn.first_name||' '||pn.last_name), (s.doctor_id||'
'||pd.first_name||' '||pd.last_name), (s.service_id ||' '||l.name)
        INTO Room, Nurse, Doctor, Service
        FROM servicerecord s, room r, nurse n, doctor d, employee en,employee ed, person pn,person pd, servicelist l
        WHERE s.room_id=r.room_id
        AND s.nurse_id=n.nurse_id AND n.nurse_id =en.employee_id AND en.employee_id = pn.person_id
        AND s.doctor_id=d.doctor_id AND d.doctor_id =ed.employee_id AND ed.employee_id = pd.person_id
        AND s.service_id=l.service_id
        AND s.room_id=r.room_id
        AND admission_id=Admission
        AND s.end_time is NULL;
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('Now:       '||SYSTIMESTAMP);
        DBMS_OUTPUT.PUT_LINE('Service:   '||Service);
        DBMS_OUTPUT.PUT_LINE('Room:      '||Room);
        DBMS_OUTPUT.PUT_LINE('Doctor:    '||Doctor);
        DBMS_OUTPUT.PUT_LINE('Nurse:     '||Nurse);
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    ELSE
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('No any current service undergo');
    END IF;

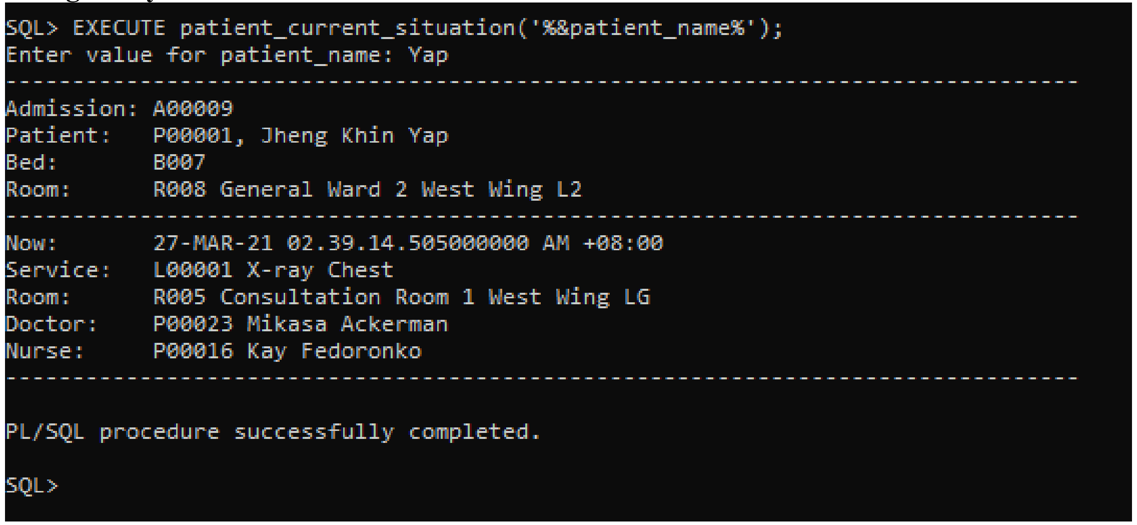
    ELSE
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('No admission_record');
    END IF;
COMMIT;
END;
/
--Execute Store Procedure
EXECUTE patient_current_situation('%&patient_name%');
```

Screenshot
Using patient name: kevin
Doing blood type test

	 ```
SQL> ---
SQL> EXECUTE patient_current_situation('%&patient_name%');
Enter value for patient_name: kevin

Admission: A00004
Patient: P00003, Kevin Owens
Bed: B004
Room: R002 General Ward 1 West Wing L2

Now: 27-MAR-21 02.38.36.689000000 AM +08:00
Service: L00004 Blood-Type Test
Room: R006 Lab West Wing LG
Doctor: P00022 Eren Yeager
Nurse: P00018 Genni Rhys

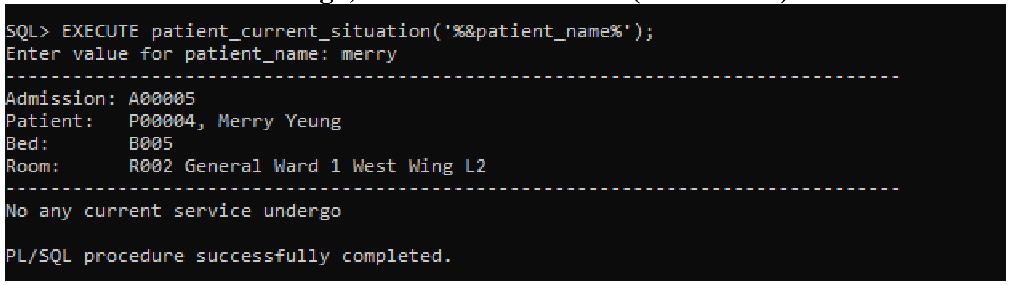
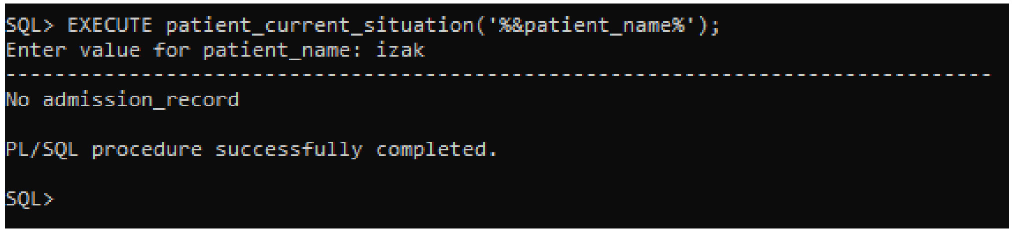
PL/SQL procedure successfully completed.
``` **Using patient name: yap** **Doing X-ray for chest at consultation room 1**  ```
SQL> EXECUTE patient_current_situation('%&patient_name%');
Enter value for patient_name: Yap

Admission: A00009
Patient: P00001, Jheng Khin Yap
Bed: B007
Room: R008 General Ward 2 West Wing L2

Now: 27-MAR-21 02.39.14.505000000 AM +08:00
Service: L00001 X-ray Chest
Room: R005 Consultation Room 1 West Wing LG
Doctor: P00023 Mikasa Ackerman
Nurse: P00016 Kay Fedoronko

PL/SQL procedure successfully completed.

SQL>
``` | |
```

	Using patient name: merry
	No current service undergo, but still in admission (Rest at bed)
	
	`SQL> EXECUTE patient_current_situation('%&patient_name%');`
	`Enter value for patient_name: merry`
	`--------------------------------------------------------------------------------`
	`Admission: A00005`
	`Patient:   P00004, Merry Yeung`
	`Bed:       B005`
	`Room:      R002 General Ward 1 West Wing L2`
	`--------------------------------------------------------------------------------`
	`No any current service undergo`
	
	`PL/SQL procedure successfully completed.`
	
	Using patient name: izak
	No admission record
	
	`SQL> EXECUTE patient_current_situation('%&patient_name%');`
	`Enter value for patient_name: izak`
	`--------------------------------------------------------------------------------`
	`No admission_record`
	
	`PL/SQL procedure successfully completed.`
	
	`SQL>`	

SP2	**<u>Insert data into medical equipment table by auto generate ID. (Add new medical equipment)</u>** • This stored procedure can insert new medical equipment by autogenerate the following medicine id. Hence, when hospital want to insert to the new medical equipment, the ID will not be mess up and provide an automation sequence followed by it. • A Trigger function on medicalequipment table and view new result(A stored procedure, view_medical_equipment) is made, when any update on medical equipment table, trigger function will execute view_medical_equipment to show the new insert row. • Lastly, there have a anonymous PL/SQL program is execute with error handling, when user wrongly input different datatype output value, it will be shown. • This insert medical equipment procedure is useful when enroll a new row without knowing the next ID in the table. This can reduce the work for the person in charge in hospital and increase his or her working efficiency. • Moreover, this procedure set up a exception case to inform user where is wrong and user can modify their input value based on the instruction given. • Lastly a trigger function is made and can show user the summary of their new insert for double confirmation.	

Stored procedure

```
--Stored Procedure - for viewing new result
CREATE OR REPLACE PROCEDURE view_medical_equipment
IS
    Medicine_count NUMBER(10);
    mname VARCHAR2(30);
    mtype VARCHAR2(12);
    mdescription VARCHAR2(50);
    mexpiration_date Date;
    munit_price Number(10,2);
    mstock_quantity Integer;
    Medicine_ID VARCHAR(6);
BEGIN
    SELECT COUNT(*) INTO Medicine_count FROM medicalequipment;

    Medicine_ID:=CONCAT('M',TRIM(TO_CHAR(Medicine_count,'09999')));

    SELECT m.name, m.type, m.description, m.expiration_date, m.unit_price, m.stock_quantity INTO mname, mtype
,mdescription,mexpiration_date, munit_price, mstock_quantity
    FROM medicalequipment m
    WHERE medicalequipment_id = Medicine_ID;

    DBMS_OUTPUT.PUT_LINE(' ID:                          '||Medicine_ID);
    DBMS_OUTPUT.PUT_LINE(' Name:                        '||mname);
    DBMS_OUTPUT.PUT_LINE(' Type:                        '||mtype);
    DBMS_OUTPUT.PUT_LINE(' Description:                 '||mdescription);
    DBMS_OUTPUT.PUT_LINE(' Expirate Date (DD/MM/YYYY): '||TO_CHAR(mexpiration_date,'DD/MM/YYYY'));
    DBMS_OUTPUT.PUT_LINE(' Unit_price:                RM'||TRIM(TO_CHAR(munit_price,'9999.99')));
    DBMS_OUTPUT.PUT_LINE(' Stock quantity:              '||TO_CHAR(mstock_quantity));
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
END;
/

--Trigger when medicine successfully insert, call the view
```

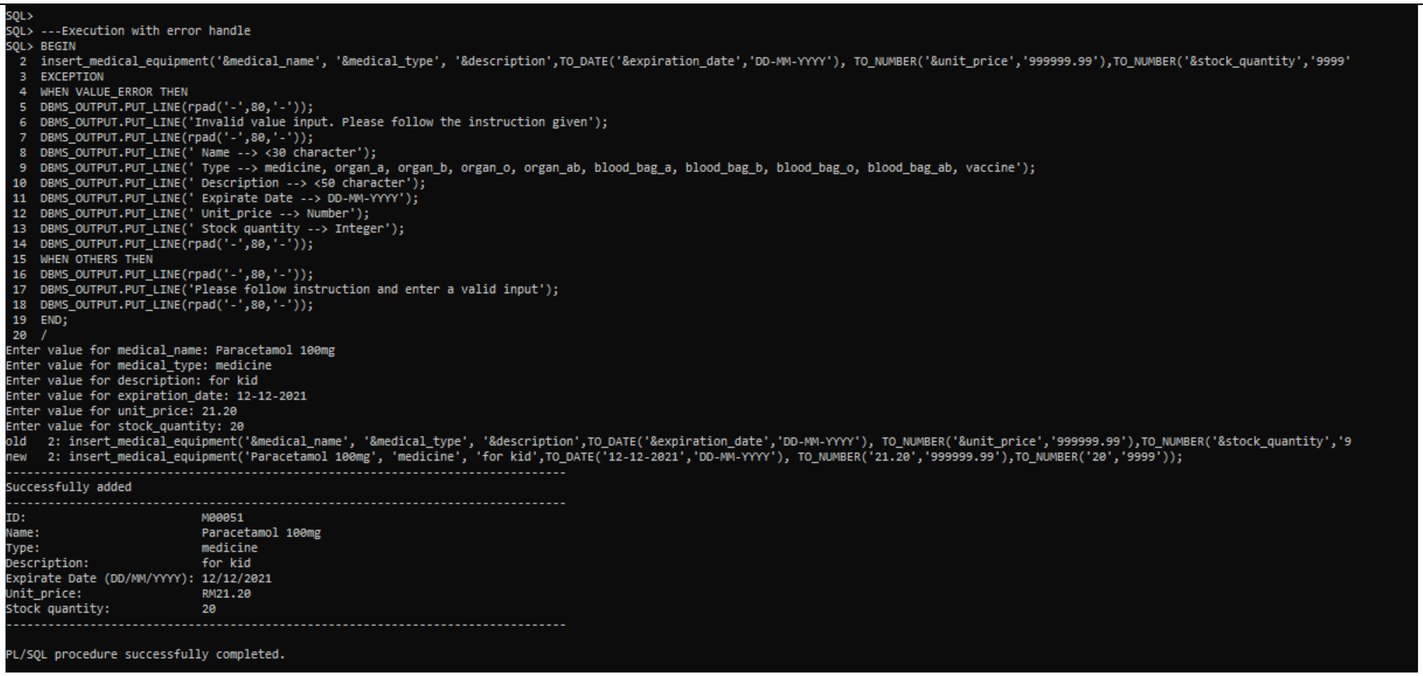
```
CREATE OR REPLACE TRIGGER insert_medicine_trigger
AFTER INSERT
  ON medicalequipment
BEGIN
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    DBMS_OUTPUT.PUT_LINE('Successfully added');
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    view_medical_equipment;
END;
/

--Stored procedure for insert medical equipment purpose (Main)
CREATE OR REPLACE PROCEDURE insert_medical_equipment
(
    name VARCHAR2,
   type VARCHAR2,
    description VARCHAR2,
   expiration_date Date,
    unit_price Number,
    stock_quantity Integer
)
IS
    Medicine_count NUMBER(10);
    Medicine_ID VARCHAR(6);
BEGIN
    SELECT COUNT(*) INTO Medicine_count FROM medicalequipment;

    Medicine_ID:=CONCAT('M',TRIM(TO_CHAR(Medicine_count+1,'09999')));

    INSERT INTO medicalequipment
    VALUES (Medicine_ID, name, type, description, expiration_date, unit_price, stock_quantity);
COMMIT;
END;
/
```

```
---Execution with error handle
BEGIN
    insert_medical_equipment('&medical_name', '&medical_type', '&description',TO_DATE('&expiration_date','DD-
MM-YYYY'), TO_NUMBER('&unit_price','999999.99'),TO_NUMBER('&stock_quantity','9999'));
EXCEPTION
    WHEN VALUE_ERROR THEN
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('Invalid value input. Please follow the instruction given');
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE(' Name --> <30 character');
        DBMS_OUTPUT.PUT_LINE(' Type --> medicine, organ_a, organ_b, organ_o, organ_ab, blood_bag_a,
blood_bag_b, blood_bag_o, blood_bag_ab, vaccine');
        DBMS_OUTPUT.PUT_LINE(' Description --> <50 character');
        DBMS_OUTPUT.PUT_LINE(' Expirate Date --> DD-MM-YYYY');
        DBMS_OUTPUT.PUT_LINE(' Unit_price --> Number');
        DBMS_OUTPUT.PUT_LINE(' Stock quantity --> Integer');
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('Please follow instruction and enter a valid input');
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
END;
/
```

<u>Screenshot</u>

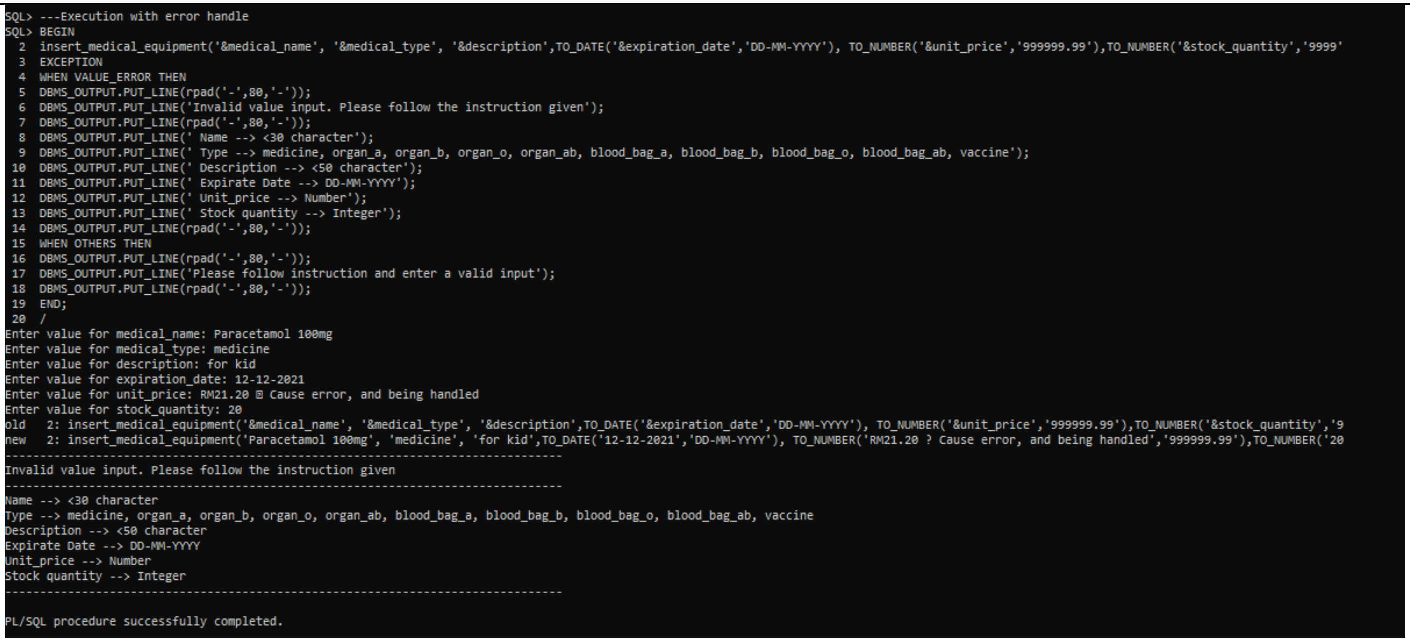
Insert using valid input (as following):

```
Paracetamol 100mg
medicine
for kid
12-12-2021
21.20
20
```



```
SQL>
SQL> ---Execution with error handle
SQL> BEGIN
  2  insert_medical_equipment('&medical_name', '&medical_type', '&description',TO_DATE('&expiration_date','DD-MM-YYYY'), TO_NUMBER('&unit_price','999999.99'),TO_NUMBER('&stock_quantity','9999'
  3  EXCEPTION
  4  WHEN VALUE_ERROR THEN
  5  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  6  DBMS_OUTPUT.PUT_LINE('Invalid value input. Please follow the instruction given');
  7  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  8  DBMS_OUTPUT.PUT_LINE(' Name --> <30 character');
  9  DBMS_OUTPUT.PUT_LINE(' Type --> medicine, organ_a, organ_b, organ_o, organ_ab, blood_bag_a, blood_bag_b, blood_bag_o, blood_bag_ab, vaccine');
 10  DBMS_OUTPUT.PUT_LINE(' Description --> <50 character');
 11  DBMS_OUTPUT.PUT_LINE(' Expirate Date --> DD-MM-YYYY');
 12  DBMS_OUTPUT.PUT_LINE(' Unit_price --> Number');
 13  DBMS_OUTPUT.PUT_LINE(' Stock quantity --> Integer');
 14  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
 15  WHEN OTHERS THEN
 16  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
 17  DBMS_OUTPUT.PUT_LINE('Please follow instruction and enter a valid input');
 18  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
 19  END;
 20  /
Enter value for medical_name: Paracetamol 100mg
Enter value for medical_type: medicine
Enter value for description: for kid
Enter value for expiration_date: 12-12-2021
Enter value for unit_price: 21.20
Enter value for stock_quantity: 20
old   2: insert_medical_equipment('&medical_name', '&medical_type', '&description',TO_DATE('&expiration_date','DD-MM-YYYY'), TO_NUMBER('&unit_price','999999.99'),TO_NUMBER('&stock_quantity','9
new   2: insert_medical_equipment('Paracetamol 100mg', 'medicine', 'for kid',TO_DATE('12-12-2021','DD-MM-YYYY'), TO_NUMBER('21.20','999999.99'),TO_NUMBER('20','9999'));
--------------------------------------------------------------------------------
Successfully added
--------------------------------------------------------------------------------
ID:                         M00051
Name:                       Paracetamol 100mg
Type:                       medicine
Description:                for kid
Expirate Date (DD/MM/YYYY): 12/12/2021
Unit_price:                 RM21.20
Stock quantity:             20
--------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

Input by invalid value (Exception case)
Paracetamol 100mg
medicine
for kid
12-12-2021
RM21.20 → Cause error, and proceed by exception
20

	
	```
	SQL> ---Execution with error handle
	SQL> BEGIN
	  2  insert_medical_equipment('&medical_name', '&medical_type', '&description',TO_DATE('&expiration_date','DD-MM-YYYY'), TO_NUMBER('&unit_price','999999.99'),TO_NUMBER('&stock_quantity','9999'
	  3  EXCEPTION
	  4  WHEN VALUE_ERROR THEN
	  5  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
	  6  DBMS_OUTPUT.PUT_LINE('Invalid value input. Please follow the instruction given');
	  7  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
	  8  DBMS_OUTPUT.PUT_LINE(' Name --> <30 character');
	  9  DBMS_OUTPUT.PUT_LINE(' Type --> medicine, organ_a, organ_b, organ_o, organ_ab, blood_bag_a, blood_bag_b, blood_bag_o, blood_bag_ab, vaccine');
	 10  DBMS_OUTPUT.PUT_LINE(' Description --> <50 character');
	 11  DBMS_OUTPUT.PUT_LINE(' Expirate Date --> DD-MM-YYYY');
	 12  DBMS_OUTPUT.PUT_LINE(' Unit_price --> Number');
	 13  DBMS_OUTPUT.PUT_LINE(' Stock quantity --> Integer');
	 14  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
	 15  WHEN OTHERS THEN
	 16  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
	 17  DBMS_OUTPUT.PUT_LINE('Please follow instruction and enter a valid input');
	 18  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
	 19  END;
	 20  /
	Enter value for medical_name: Paracetamol 100mg
	Enter value for medical_type: medicine
	Enter value for description: for kid
	Enter value for expiration_date: 12-12-2021
	Enter value for unit_price: RM21.20 ⊠ Cause error, and being handled
	Enter value for stock_quantity: 20
	old   2: insert_medical_equipment('&medical_name', '&medical_type', '&description',TO_DATE('&expiration_date','DD-MM-YYYY'), TO_NUMBER('&unit_price','999999.99'),TO_NUMBER('&stock_quantity','9
	new   2: insert_medical_equipment('Paracetamol 100mg', 'medicine', 'for kid',TO_DATE('12-12-2021','DD-MM-YYYY'), TO_NUMBER('RM21.20 ? Cause error, and being handled','999999.99'),TO_NUMBER('20
	--------------------------------------------------------------------------------
	Invalid value input. Please follow the instruction given
	--------------------------------------------------------------------------------
	Name --> <30 character
	Type --> medicine, organ_a, organ_b, organ_o, organ_ab, blood_bag_a, blood_bag_b, blood_bag_o, blood_bag_ab, vaccine
	Description --> <50 character
	Expirate Date --> DD-MM-YYYY
	Unit_price --> Number
	Stock quantity --> Integer
	--------------------------------------------------------------------------------

	PL/SQL procedure successfully completed.
	```

SP3	**<u>Update salary of all staff in hospital with a parameter value and option procedure</u>** • The Hospital need to update patient salary due to certain circumstances including but not limited to ‘annual increment’, ‘financial bottleneck decrement’. • The stored procedure is helpful for modification large amount of salary data. • This stored procedure provides parameter (which the parameter value) and the option (which is the function operation) For instance, when parameter is 200, function is ‘+’ mean +200 to all employee. For instance, when parameter is 100, function is ‘-’ mean -100 to all employee. For instance, when parameter is 50, function is ‘*’ mean 50% of employee salary is decreased. For instance, when parameter is 150, function is ‘*’ mean 50% increment of employee salary. **<u>Stored Procedure</u>** CREATE OR REPLACE PROCEDURE update_salary (uparameter Number,	

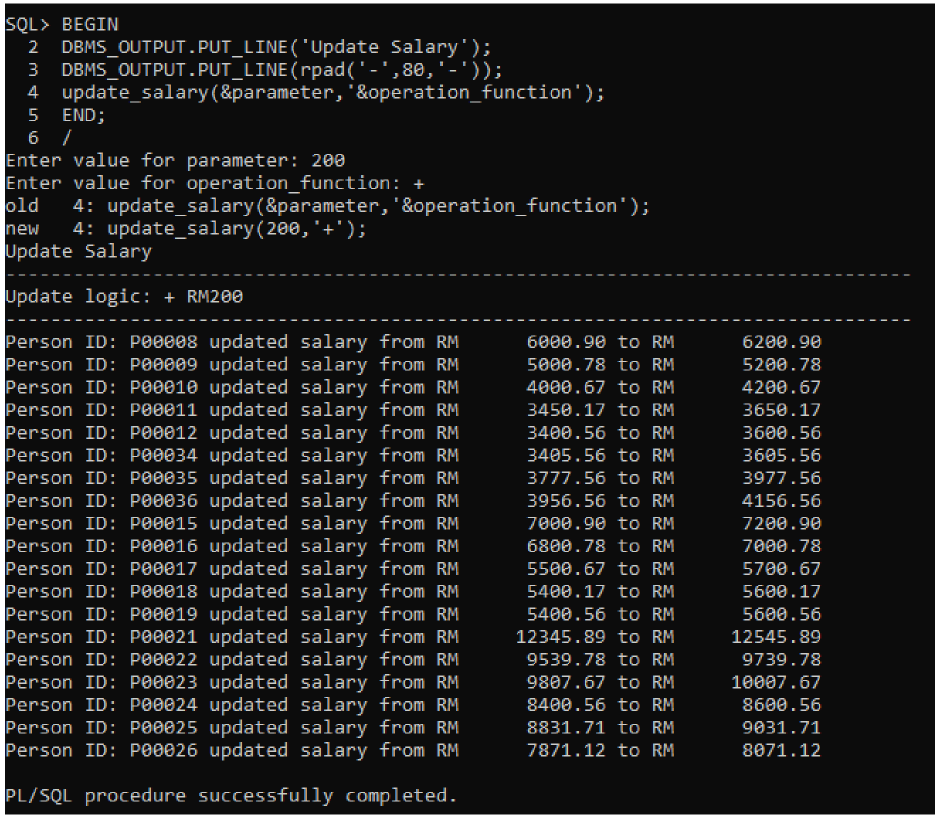
```
    ufunction VARCHAR
)
IS
    pid employee.employee_id%type;
    pidc employee.employee_id%type;
    psalary employee.salary%type;
    pleave_date employee.leave_date%type;
    CURSOR pointer is
        SELECT employee_id,salary,leave_date
        FROM employee;
BEGIN

        IF ufunction='+' THEN
            DBMS_OUTPUT.PUT_LINE('Update logic: + RM'||uparameter);
        ELSIF ufunction='-' THEN
            DBMS_OUTPUT.PUT_LINE('Update logic: - RM'||uparameter);
        ELSIF ufunction='*' THEN
            DBMS_OUTPUT.PUT_LINE('Update logic: * '||TO_CHAR(uparameter)||'%');
        ELSE
            DBMS_OUTPUT.PUT_LINE('Invalid input. Only accept + - and *');
            RETURN;
        END IF;

        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));

        OPEN pointer;
        LOOP
        FETCH pointer INTO pid, psalary, pleave_date;
        IF pidc=pid THEN
            EXIT;
        ELSE
        pidc:=pid;
        END IF;
        IF pleave_date IS NULL THEN
            IF ufunction='+' THEN
        UPDATE employee set salary=psalary+uparameter WHERE employee_id=pid;
        DBMS_OUTPUT.PUT_LINE('Person ID: '||TO_CHAR(pid)||' updated salary from
RM'||TO_CHAR(psalary,'999999999.99')||' to RM'||TO_CHAR(psalary+uparameter,'999999999.99'));
      ELSIF ufunction='-' THEN
        UPDATE employee set salary=psalary-uparameter WHERE employee_id=pid;
        DBMS_OUTPUT.PUT_LINE('Person ID: '||TO_CHAR(pid)||' updated salary from
RM'||TO_CHAR(psalary,'999999999.99')||' to RM'||TO_CHAR(psalary-uparameter,'999999999.99'));
      ELSIF ufunction='*' THEN
        UPDATE employee set salary=psalary/100*uparameter WHERE employee_id=pid;
        DBMS_OUTPUT.PUT_LINE('Person ID: '||TO_CHAR(pid)||' updated salary from
RM'||TO_CHAR(psalary,'999999999.99')||' to RM'||TO_CHAR(psalary/100*uparameter,'999999999.99'));
      END IF;
    END IF;
    EXIT WHEN pointer%NOTFOUND;

    END LOOP;
    CLOSE pointer;
COMMIT;
END;
/

--Execution
BEGIN
DBMS_OUTPUT.PUT_LINE('Update Salary');
DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
update_salary(&parameter,'&operation_function');
END;
/
```

Screenshot

Insert by valid input (parameter and function option)

200

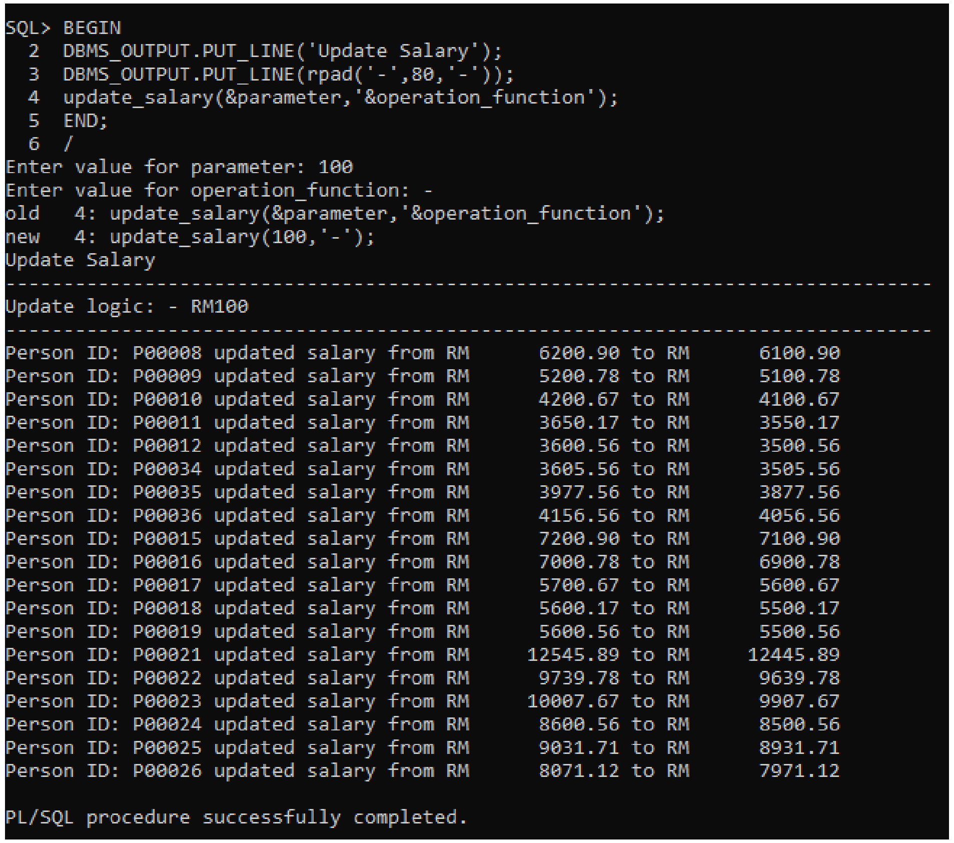
+



```
SQL> BEGIN
  2  DBMS_OUTPUT.PUT_LINE('Update Salary');
  3  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  4  update_salary(&parameter,'&operation_function');
  5  END;
  6  /
Enter value for parameter: 200
Enter value for operation_function: +
old   4: update_salary(&parameter,'&operation_function');
new   4: update_salary(200,'+');
Update Salary
--------------------------------------------------------------------------------
Update logic: + RM200
--------------------------------------------------------------------------------
Person ID: P00008 updated salary from RM      6000.90 to RM      6200.90
Person ID: P00009 updated salary from RM      5000.78 to RM      5200.78
Person ID: P00010 updated salary from RM      4000.67 to RM      4200.67
Person ID: P00011 updated salary from RM      3450.17 to RM      3650.17
Person ID: P00012 updated salary from RM      3400.56 to RM      3600.56
Person ID: P00034 updated salary from RM      3405.56 to RM      3605.56
Person ID: P00035 updated salary from RM      3777.56 to RM      3977.56
Person ID: P00036 updated salary from RM      3956.56 to RM      4156.56
Person ID: P00015 updated salary from RM      7000.90 to RM      7200.90
Person ID: P00016 updated salary from RM      6800.78 to RM      7000.78
Person ID: P00017 updated salary from RM      5500.67 to RM      5700.67
Person ID: P00018 updated salary from RM      5400.17 to RM      5600.17
Person ID: P00019 updated salary from RM      5400.56 to RM      5600.56
Person ID: P00021 updated salary from RM     12345.89 to RM     12545.89
Person ID: P00022 updated salary from RM      9539.78 to RM      9739.78
Person ID: P00023 updated salary from RM      9807.67 to RM     10007.67
Person ID: P00024 updated salary from RM      8400.56 to RM      8600.56
Person ID: P00025 updated salary from RM      8831.71 to RM      9031.71
Person ID: P00026 updated salary from RM      7871.12 to RM      8071.12

PL/SQL procedure successfully completed.
```

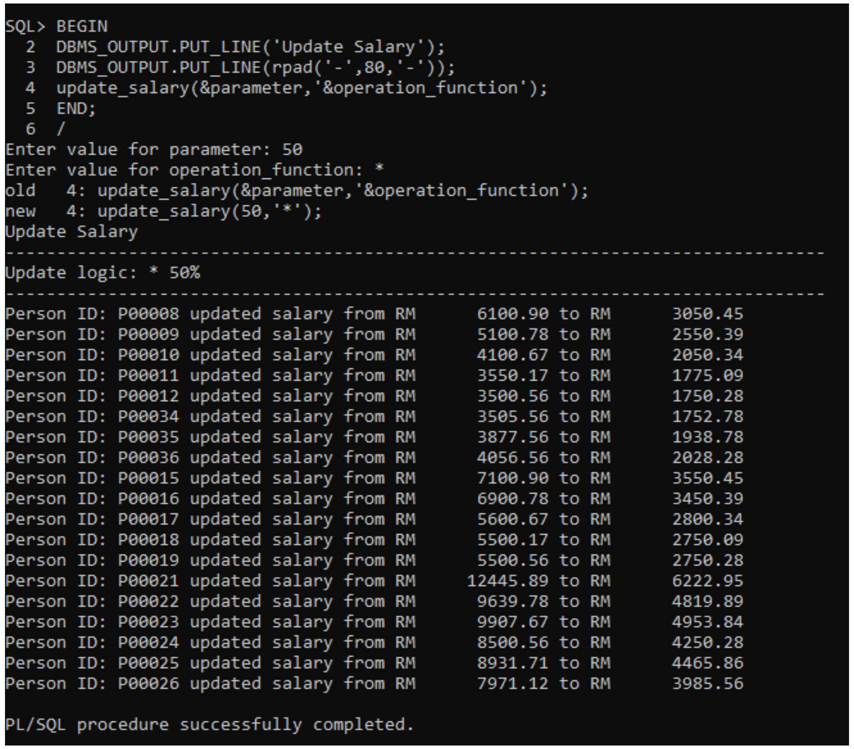
100

-



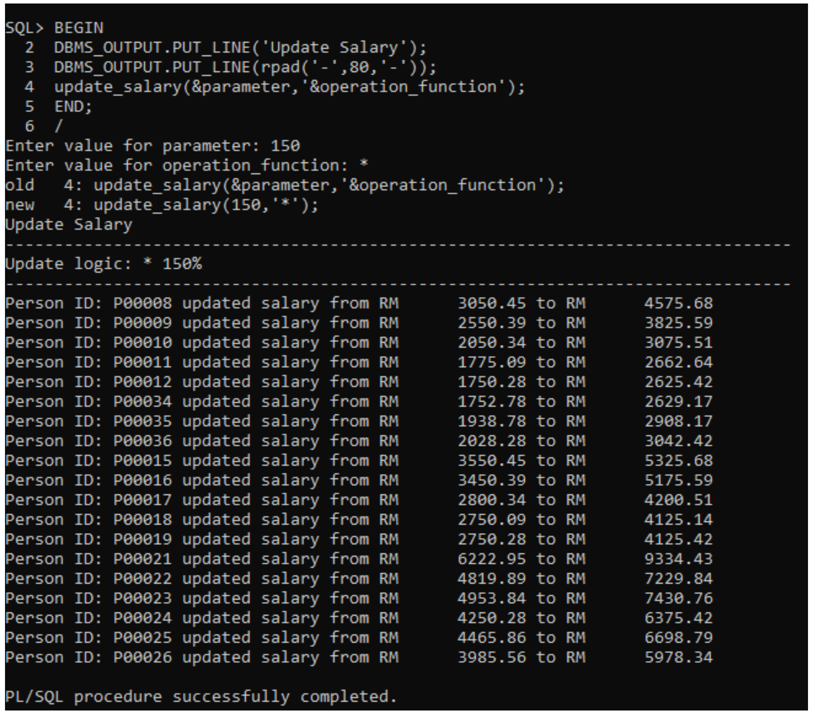
```
SQL> BEGIN
  2  DBMS_OUTPUT.PUT_LINE('Update Salary');
  3  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  4  update_salary(&parameter,'&operation_function');
  5  END;
  6  /
Enter value for parameter: 100
Enter value for operation_function: -
old   4: update_salary(&parameter,'&operation_function');
new   4: update_salary(100,'-');
Update Salary
--------------------------------------------------------------------------------
Update logic: - RM100
--------------------------------------------------------------------------------
Person ID: P00008 updated salary from RM      6200.90 to RM      6100.90
Person ID: P00009 updated salary from RM      5200.78 to RM      5100.78
Person ID: P00010 updated salary from RM      4200.67 to RM      4100.67
Person ID: P00011 updated salary from RM      3650.17 to RM      3550.17
Person ID: P00012 updated salary from RM      3600.56 to RM      3500.56
Person ID: P00034 updated salary from RM      3605.56 to RM      3505.56
Person ID: P00035 updated salary from RM      3977.56 to RM      3877.56
Person ID: P00036 updated salary from RM      4156.56 to RM      4056.56
Person ID: P00015 updated salary from RM      7200.90 to RM      7100.90
Person ID: P00016 updated salary from RM      7000.78 to RM      6900.78
Person ID: P00017 updated salary from RM      5700.67 to RM      5600.67
Person ID: P00018 updated salary from RM      5600.17 to RM      5500.17
Person ID: P00019 updated salary from RM      5600.56 to RM      5500.56
Person ID: P00021 updated salary from RM     12545.89 to RM     12445.89
Person ID: P00022 updated salary from RM      9739.78 to RM      9639.78
Person ID: P00023 updated salary from RM     10007.67 to RM      9907.67
Person ID: P00024 updated salary from RM      8600.56 to RM      8500.56
Person ID: P00025 updated salary from RM      9031.71 to RM      8931.71
Person ID: P00026 updated salary from RM      8071.12 to RM      7971.12

PL/SQL procedure successfully completed.
```

50
*



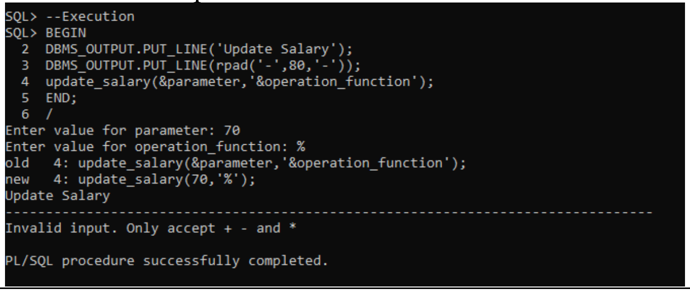
```
SQL> BEGIN
  2  DBMS_OUTPUT.PUT_LINE('Update Salary');
  3  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  4  update_salary(&parameter,'&operation_function');
  5  END;
  6  /
Enter value for parameter: 50
Enter value for operation_function: *
old   4: update_salary(&parameter,'&operation_function');
new   4: update_salary(50,'*');
Update Salary
--------------------------------------------------------------------------------
Update logic: * 50%
--------------------------------------------------------------------------------
Person ID: P00008 updated salary from RM      6100.90 to RM      3050.45
Person ID: P00009 updated salary from RM      5100.78 to RM      2550.39
Person ID: P00010 updated salary from RM      4100.67 to RM      2050.34
Person ID: P00011 updated salary from RM      3550.17 to RM      1775.09
Person ID: P00012 updated salary from RM      3500.56 to RM      1750.28
Person ID: P00034 updated salary from RM      3505.56 to RM      1752.78
Person ID: P00035 updated salary from RM      3877.56 to RM      1938.78
Person ID: P00036 updated salary from RM      4056.56 to RM      2028.28
Person ID: P00015 updated salary from RM      7100.90 to RM      3550.45
Person ID: P00016 updated salary from RM      6900.78 to RM      3450.39
Person ID: P00017 updated salary from RM      5600.67 to RM      2800.34
Person ID: P00018 updated salary from RM      5500.17 to RM      2750.09
Person ID: P00019 updated salary from RM      5500.56 to RM      2750.28
Person ID: P00021 updated salary from RM     12445.89 to RM      6222.95
Person ID: P00022 updated salary from RM      9639.78 to RM      4819.89
Person ID: P00023 updated salary from RM      9907.67 to RM      4953.84
Person ID: P00024 updated salary from RM      8500.56 to RM      4250.28
Person ID: P00025 updated salary from RM      8931.71 to RM      4465.86
Person ID: P00026 updated salary from RM      7971.12 to RM      3985.56

PL/SQL procedure successfully completed.
```

150
*



```
SQL> BEGIN
  2  DBMS_OUTPUT.PUT_LINE('Update Salary');
  3  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  4  update_salary(&parameter,'&operation_function');
  5  END;
  6  /
Enter value for parameter: 150
Enter value for operation_function: *
old   4: update_salary(&parameter,'&operation_function');
new   4: update_salary(150,'*');
Update Salary
--------------------------------------------------------------------------------
Update logic: * 150%
--------------------------------------------------------------------------------
Person ID: P00008 updated salary from RM      3050.45 to RM      4575.68
Person ID: P00009 updated salary from RM      2550.39 to RM      3825.59
Person ID: P00010 updated salary from RM      2050.34 to RM      3075.51
Person ID: P00011 updated salary from RM      1775.09 to RM      2662.64
Person ID: P00012 updated salary from RM      1750.28 to RM      2625.42
Person ID: P00034 updated salary from RM      1752.78 to RM      2629.17
Person ID: P00035 updated salary from RM      1938.78 to RM      2908.17
Person ID: P00036 updated salary from RM      2028.28 to RM      3042.42
Person ID: P00015 updated salary from RM      3550.45 to RM      5325.68
Person ID: P00016 updated salary from RM      3450.39 to RM      5175.59
Person ID: P00017 updated salary from RM      2800.34 to RM      4200.51
Person ID: P00018 updated salary from RM      2750.09 to RM      4125.14
Person ID: P00019 updated salary from RM      2750.28 to RM      4125.42
Person ID: P00021 updated salary from RM      6222.95 to RM      9334.43
Person ID: P00022 updated salary from RM      4819.89 to RM      7229.84
Person ID: P00023 updated salary from RM      4953.84 to RM      7430.76
Person ID: P00024 updated salary from RM      4250.28 to RM      6375.42
Person ID: P00025 updated salary from RM      4465.86 to RM      6698.79
Person ID: P00026 updated salary from RM      3985.56 to RM      5978.34

PL/SQL procedure successfully completed.
```

Invalid input handle
70
% → Invalid inputs



```
SQL> --Execution
SQL> BEGIN
  2  DBMS_OUTPUT.PUT_LINE('Update Salary');
  3  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  4  update_salary(&parameter,'&operation_function');
  5  END;
  6  /
Enter value for parameter: 70
Enter value for operation_function: %
old   4: update_salary(&parameter,'&operation_function');
new   4: update_salary(70,'%');
Update Salary
--------------------------------------------------------------------------------
Invalid input. Only accept + - and *

PL/SQL procedure successfully completed.
```

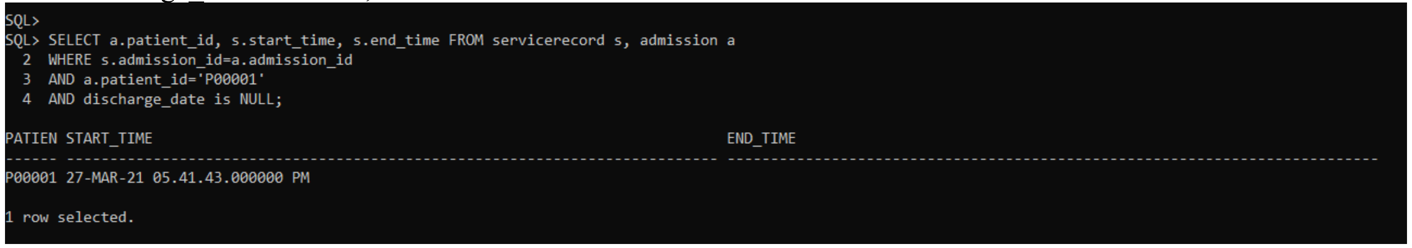
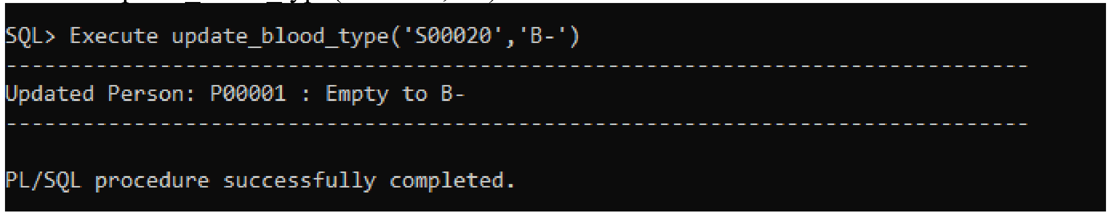
SP4

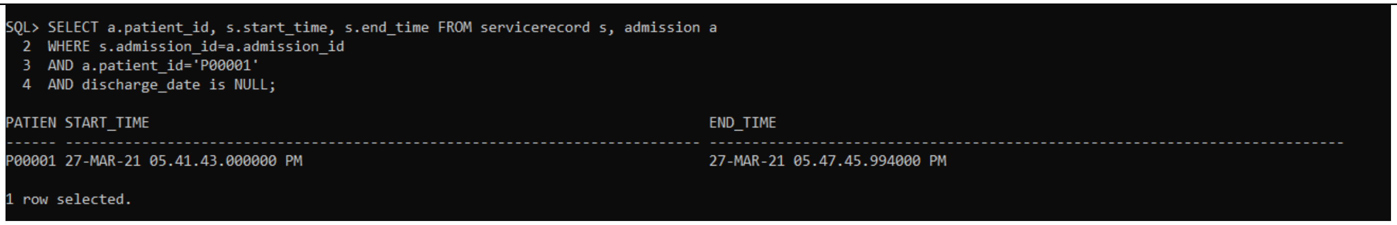
<u>Update blood type in both person table and servicerecord table</u>

- The Hospital need to update patient blood type if he or she undergo a blood type testing in hospital
- A stored procedure is write for update blood type.
- A trigger had been made for any person update. It will shown the This is vital to show the historical of patient's blood type as blood type is a very important reference in medical field.
- A call function for lab doctor to update patient lab record is shown

<u>Stored Procedure</u>

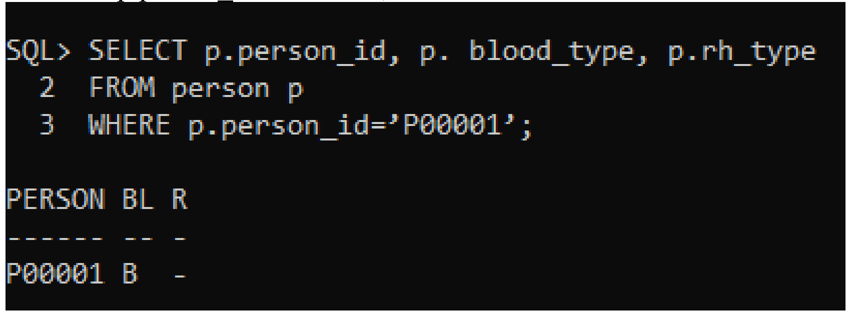
```
CREATE OR REPLACE PROCEDURE update_blood_type(
    nservice_record_id VARCHAR,
    nsummary VARCHAR
)
IS
    bt VARCHAR(2);
    rh VARCHAR(1);
    pid VARCHAR(6);
BEGIN
    SELECT p.patient_id into pid
    FROM servicerecord s, admission a, patient p
    WHERE s.admission_id = a.admission_id
    AND a.patient_id=p.patient_id
    AND s.service_record_id=nservice_record_id;

    IF LENGTH(nsummary)=3 THEN
        bt:=SUBSTR(nsummary,1,2);
        rh:=SUBSTR(nsummary,3,1);
    ELSIF LENGTH(nsummary)=2 THEN
        bt:=SUBSTR(nsummary,1,1);
        rh:=SUBSTR(nsummary,2,1);
    END IF;

    UPDATE person p set p.blood_type = bt,p.rh_type=rh WHERE p.person_id= pid;
    UPDATE servicerecord s set s.summary=nsummary, s.end_time=SYSTIMESTAMP WHERE
s.service_record_id=nservice_record_id;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Invalid input. Please follow the instruction given');
END;
/

--Trigger to view comparison
CREATE OR REPLACE TRIGGER trigger_patient_blood_type
AFTER UPDATE
  ON person
FOR EACH ROW
BEGIN
IF(:old.blood_type = :new.blood_type) AND (:old.rh_type = :new.rh_type) THEN
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    DBMS_OUTPUT.PUT_LINE('No blood type is changed. Blood type: '||:new.blood_type||:new.rh_type);
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
ELSE
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    DBMS_OUTPUT.PUT_LINE('Updated Person: '||:old.person_id||' : '||CASE WHEN :old.blood_type is NULL OR
:old.rh_type is NULL THEN 'Empty' ELSE CONCAT(:old.blood_type,:old.rh_type) END||' to
'||:new.blood_type||:new.rh_type);
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
END IF;
END;
/

--Execution
Execute update_blood_type('S00020','B-')
```

	<u>Screenshot</u>
	
	First, we can see there are a current service for P00001 patient
	SELECT a.patient_id, s.start_time, s.end_time FROM servicerecord s, admission a
	WHERE s.admission_id=a.admission_id
	AND a.patient_id='P00001'
	AND discharge_date is NULL;
	
	`SQL>`
	`SQL> SELECT a.patient_id, s.start_time, s.end_time FROM servicerecord s, admission a`
	`  2  WHERE s.admission_id=a.admission_id`
	`  3  AND a.patient_id='P00001'`
	`  4  AND discharge_date is NULL;`
	
	`PATIEN START_TIME                                                                  END_TIME`
	`------ --------------------------------------------------------------------------- ---------------------------------------------------------------------------`
	`P00001 27-MAR-21 05.41.43.000000 PM`
	
	`1 row selected.`
	
	Lab doctor update the blood with this stored procedure. A trigger is shown the different between before and after. This is just make sure incase any contradiction for history.
	Execute update_blood_type('S00020','B-')
	
	`SQL> Execute update_blood_type('S00020','B-')`
	`--------------------------------------------------------------------------------`
	`Updated Person: P00001 : Empty to B-`
	`--------------------------------------------------------------------------------`
	
	`PL/SQL procedure successfully completed.`
	
	When execute the upper stored procedure, blood test result will input in service record summary and automatically update to patient record. We can verify servicerecord table is updated as shown end time
	SELECT a.patient_id, s.start_time, s.end_time FROM servicerecord s, admission a
	WHERE s.admission_id=a.admission_id
	AND a.patient_id='P00001'
	AND discharge_date is NULL;	



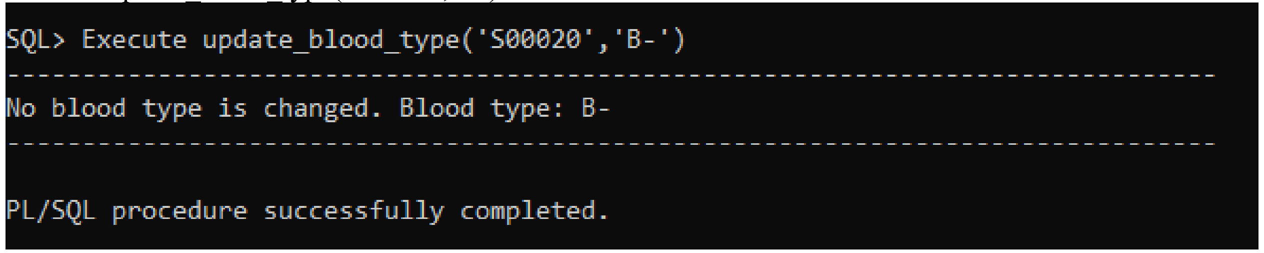
```
SQL> SELECT a.patient_id, s.start_time, s.end_time FROM servicerecord s, admission a
WHERE s.admission_id=a.admission_id
AND a.patient_id='P00001'
AND discharge_date is NULL;

PATIEN START_TIME                                                                   END_TIME
------ ---------------------------------------------------------------------------- ---------------------------------------------------------------------------
P00001 27-MAR-21 05.41.43.000000 PM                                                 27-MAR-21 05.47.45.994000 PM

1 row selected.
```

Verify person record
SELECT p.person_id, p. blood_type, p.rh_type
FROM person p
WHERE p.person_id=’P00001’;



```
SQL> SELECT p.person_id, p. blood_type, p.rh_type
FROM person p
WHERE p.person_id=’P00001’;

PERSON BL R
------ -- -
P00001 B  -
```

When further update incurs, the trigger function will execute to view the different between before and after.
Execute update_blood_type('S00020','B-')



```
SQL> Execute update_blood_type('S00020','B-')
--------------------------------------------------------------------------
No blood type is changed. Blood type: B-
--------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

SP5

Update all unpaid overdue bill’s due date and adding penalty.

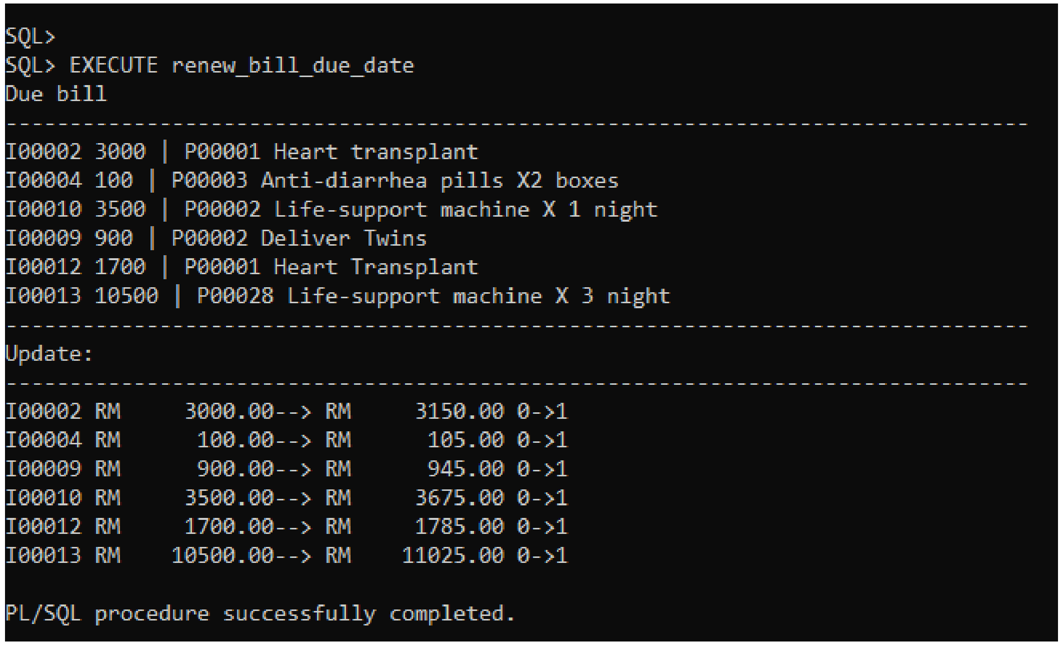
- A call function for Hospital staff to update the due date of bill and incurring charge for those have not pay after due date
- This will increase staff efficiency when they update the overdue bill
- Nested call of procedure is used to reach the intention result.
- The first stored procedure is to list out all unpaid and overdue bill
- The second store procedure function as update the overdue bill and calling first stored procedure to show the result

Stored Procedure

```
-- Stored procedure - show unpaid
CREATE OR REPLACE PROCEDURE show_unpaid
IS
    CURSOR pointers is
        SELECT b.bill_id AS ID, b.amount AS Amount,b.description AS Description, pt.patient_id
        FROM patient pt, person p, bill b, admission a
        WHERE b.admission_id=a.admission_id
        AND a.patient_id=pt.patient_id
        AND pt.patient_id=p.person_id
        AND b.payment_date is null
        AND TRUNC(SYSDATE-b.due_date)>0;
BEGIN
        DBMS_OUTPUT.PUT_LINE('Due bill');
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        FOR ptr IN pointers
        LOOP
            DBMS_OUTPUT.PUT_LINE(ptr.id||' '||ptr.amount||' | '||ptr.patient_id||' '|| ptr.description);
        END LOOP;
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
COMMIT;
END;
/
-- Stored procedure (main)- update
CREATE OR REPLACE PROCEDURE renew_bill_due_date
IS
    times Number;
    counti Number;
    temp VARCHAR(70);
    CURSOR pointers is
        SELECT bill_id,due_date,amount,description,payment_date FROM bill;
BEGIN
        show_unpaid;
        DBMS_OUTPUT.PUT_LINE('Update:');
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        FOR ptr IN pointers
        LOOP
            IF ptr.payment_date IS NULL AND SYSDATE>ptr.due_date THEN
                counti:=INSTR(ptr.description,'*');
                IF counti>0 THEN
                    counti:=INSTR(ptr.description,'*');
                    times:=TO_NUMBER(SUBSTR(ptr.description,0,counti));
                    temp:=SUBSTR(ptr.description,counti,60);
                ELSE
                    temp:=CONCAT('*',ptr.description);
                    times:=0;
                END IF;
                times:=times+1;
                temp:=SUBSTR(CONCAT(times,temp),1,60);
                UPDATE bill SET due_date=SYSDATE+14, amount=ptr.amount*105/100, description=temp WHERE
bill_id=ptr.bill_id;
                DBMS_OUTPUT.PUT_LINE(ptr.bill_id||' RM'||TO_CHAR(ptr.amount,'99999999.99')||'-->
RM'||TO_CHAR(ptr.amount*105/100,'99999999.99')||' '||TO_CHAR(times-1)||'->'||TO_CHAR(times));
            END IF;
        END LOOP;
COMMIT;
END;
/

--Execution
EXECUTE renew_bill_due_date
```

Screenshot

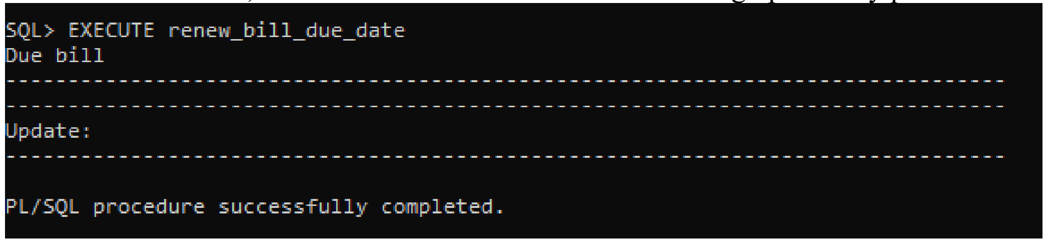
The normal execution



```
SQL>
SQL> EXECUTE renew_bill_due_date
Due bill
--------------------------------------------------------------------------------
I00002 3000 | P00001 Heart transplant
I00004 100 | P00003 Anti-diarrhea pills X2 boxes
I00010 3500 | P00002 Life-support machine X 1 night
I00009 900 | P00002 Deliver Twins
I00012 1700 | P00001 Heart Transplant
I00013 10500 | P00028 Life-support machine X 3 night
--------------------------------------------------------------------------------
Update:
--------------------------------------------------------------------------------
I00002 RM     3000.00--> RM     3150.00 0->1
I00004 RM      100.00--> RM      105.00 0->1
I00009 RM      900.00--> RM      945.00 0->1
I00010 RM     3500.00--> RM     3675.00 0->1
I00012 RM     1700.00--> RM     1785.00 0->1
I00013 RM    10500.00--> RM    11025.00 0->1

PL/SQL procedure successfully completed.
```

Validation

Execution two times, no overdue bill is shown as all of it being updated by previous execution



```
SQL> EXECUTE renew_bill_due_date
Due bill
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Update:
--------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

Function (10 marks)		
F1	**Calculate and list out total available bed in all room or specify room**	

- List out all the available bed that current have in hospital
- This is helping the nurse assign patient to respective room and bed.
- This function also useful for doing analysis of current available bed. if there have not sufficient bed, hospital can prepare to buy more bed or prepare for hospital transfer
- This function can receive two type of input. One is room_id and another is ALL to show all available bed and room

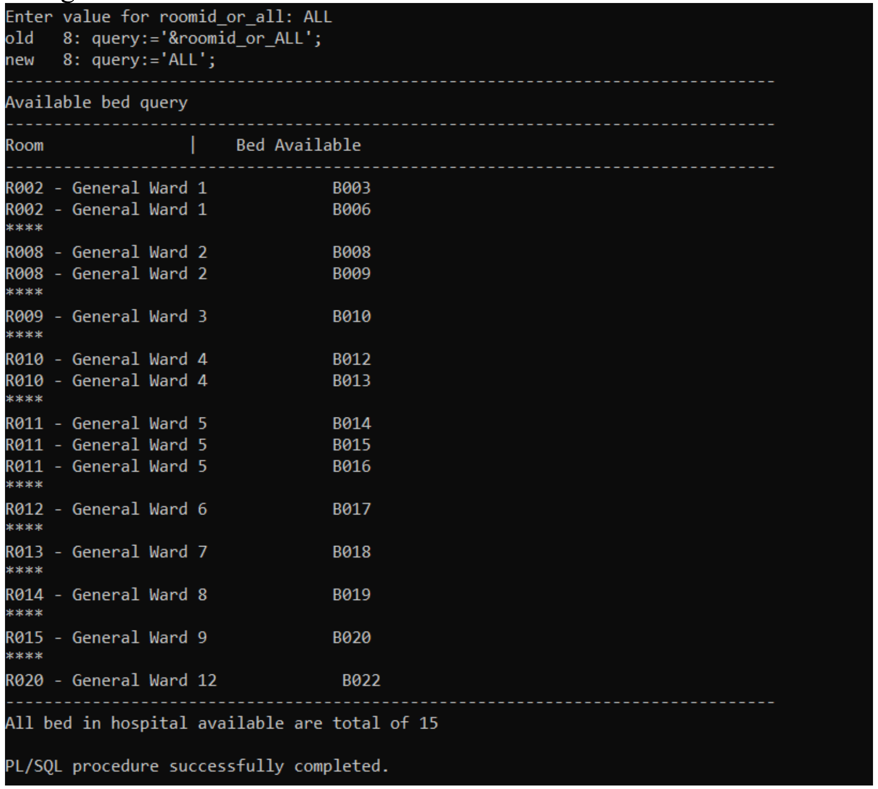
Function

```
--Function
CREATE OR REPLACE FUNCTION total_bed_available
(
    roomid VARCHAR
)
RETURN NUMBER
IS
bedcount number(8);
temp VARCHAR(4);
CURSOR pointers is
    SELECT b.bed_id AS Bed, r.room_id AS Room, r.room_name AS RName
    FROM bed b, room r
    WHERE b.room_id=r.room_id
    MINUS
    SELECT b.bed_id AS Bed, r.room_id AS Room, r.room_name AS RName
    FROM admission a, bed b, room r
    WHERE a.bed_id = b.bed_id
    AND b.room_id=r.room_id
    AND a.discharge_date IS NULL
    AND a.status IN ('I','R');
BEGIN
    bedcount:=0;
    temp:=' ';
    DBMS_OUTPUT.PUT_LINE('Room              |   Bed Available');
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    FOR ptr IN pointers
    LOOP
    IF UPPER(roomid) = 'ALL' THEN
        IF temp=' ' THEN
            DBMS_OUTPUT.PUT_LINE(' ');
        ELSIF temp!=ptr.Room THEN
            DBMS_OUTPUT.PUT_LINE('****');
        END IF;
        temp:=ptr.Room;
        bedcount:=bedcount+1;
        DBMS_OUTPUT.PUT_LINE(ptr.Room||' - '||ptr.RName||'            '||ptr.Bed);
    ELSIF roomid = ptr.Room THEN
        bedcount:=bedcount+1;
        DBMS_OUTPUT.PUT_LINE(ptr.Room||' - '||ptr.RName||'            '||ptr.Bed);
    END IF;
    END LOOP;
    RETURN bedcount;
END;
/
--Execution
DECLARE
    totalbed number(8);
    query VARCHAR(6);
BEGIN
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    DBMS_OUTPUT.PUT_LINE('Available bed query');
    DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    query:='&roomid_or_ALL';
    IF UPPER(query)='ALL' THEN
        totalbed:=total_bed_available(query);
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('All bed in hospital available are total of '||TO_NUMBER(totalbed,'9999'));
    ELSIF REGEXP_LIKE(query,'^R\d{3}$') THEN
        totalbed:=total_bed_available(query);
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('All bed in room with ID '||query||' have total of '||TO_NUMBER(totalbed,'9999'));
    ELSE
        DBMS_OUTPUT.PUT_LINE('Invalid input. Key in again');
END IF;
END;
/
```

<u>Screenshot</u>

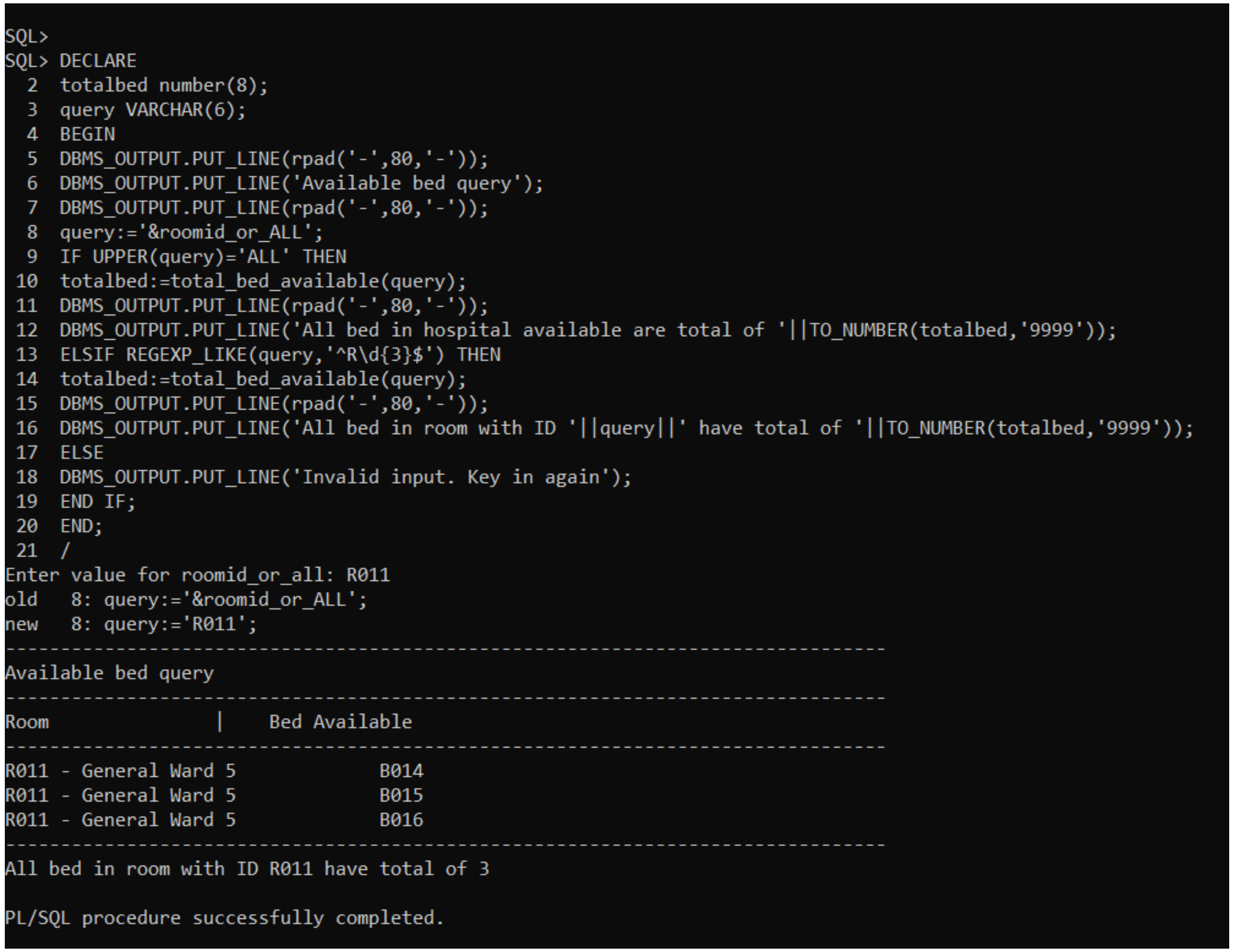
Calculate and show the list of available bed in all room

Using: ALL



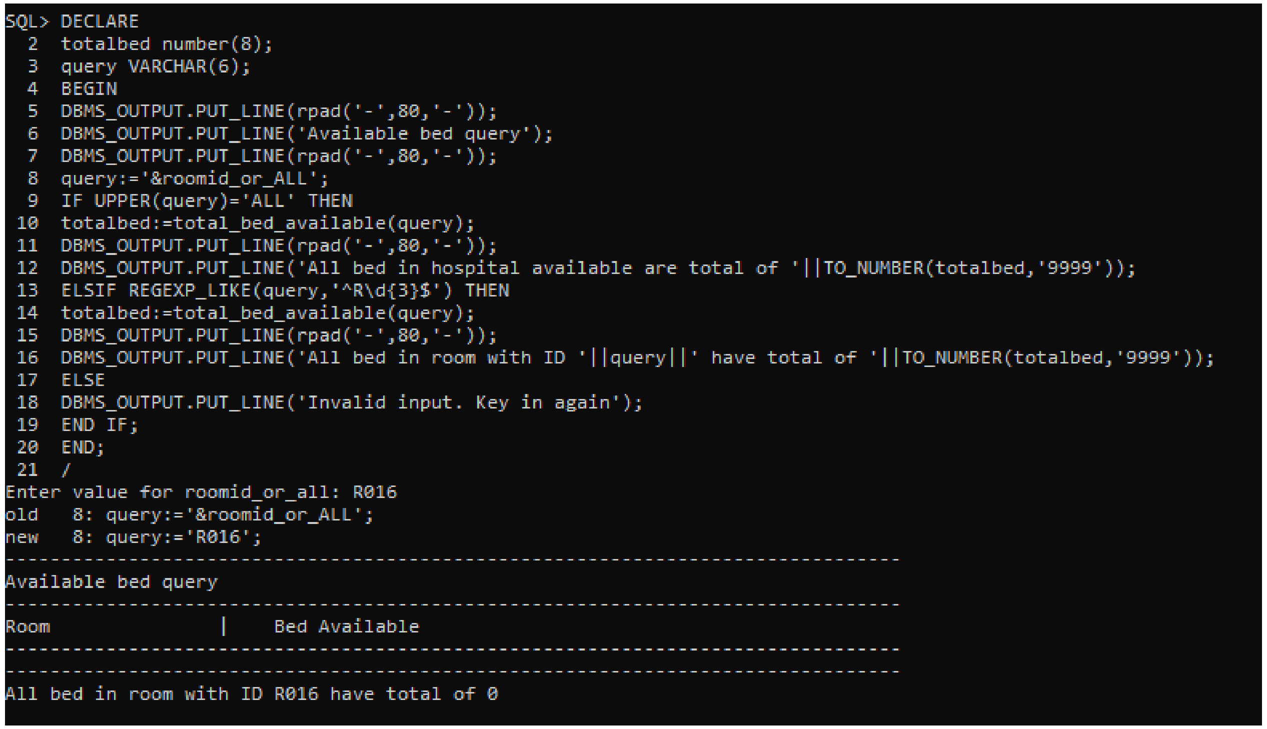
```
Enter value for roomid_or_all: ALL
old   8: query:='&roomid_or_ALL';
new   8: query:='ALL';
--------------------------------------------------------------------------------
Available bed query
--------------------------------------------------------------------------------
Room              |    Bed Available
--------------------------------------------------------------------------------
R002 - General Ward 1            B003
R002 - General Ward 1            B006
****
R008 - General Ward 2            B008
R008 - General Ward 2            B009
****
R009 - General Ward 3            B010
****
R010 - General Ward 4            B012
R010 - General Ward 4            B013
****
R011 - General Ward 5            B014
R011 - General Ward 5            B015
R011 - General Ward 5            B016
****
R012 - General Ward 6            B017
****
R013 - General Ward 7            B018
****
R014 - General Ward 8            B019
****
R015 - General Ward 9            B020
****
R020 - General Ward 12            B022
--------------------------------------------------------------------------------
All bed in hospital available are total of 15

PL/SQL procedure successfully completed.
```

Calculate and show the list of available bed in particular room
Using: R011



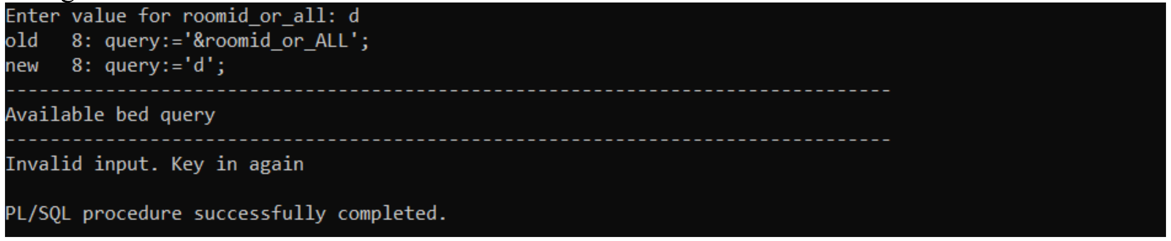
```
SQL>
SQL> DECLARE
  2  totalbed number(8);
  3  query VARCHAR(6);
  4  BEGIN
  5  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  6  DBMS_OUTPUT.PUT_LINE('Available bed query');
  7  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  8  query:='&roomid_or_ALL';
  9  IF UPPER(query)='ALL' THEN
 10  totalbed:=total_bed_available(query);
 11  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
 12  DBMS_OUTPUT.PUT_LINE('All bed in hospital available are total of '||TO_NUMBER(totalbed,'9999'));
 13  ELSIF REGEXP_LIKE(query,'^R\d{3}$') THEN
 14  totalbed:=total_bed_available(query);
 15  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
 16  DBMS_OUTPUT.PUT_LINE('All bed in room with ID '||query||' have total of '||TO_NUMBER(totalbed,'9999'));
 17  ELSE
 18  DBMS_OUTPUT.PUT_LINE('Invalid input. Key in again');
 19  END IF;
 20  END;
 21  /
Enter value for roomid_or_all: R011
old   8: query:='&roomid_or_ALL';
new   8: query:='R011';
--------------------------------------------------------------------------------
Available bed query
--------------------------------------------------------------------------------
Room              |    Bed Available
--------------------------------------------------------------------------------
R011 - General Ward 5            B014
R011 - General Ward 5            B015
R011 - General Ward 5            B016
--------------------------------------------------------------------------------
All bed in room with ID R011 have total of 3

PL/SQL procedure successfully completed.
```

Calculate and show the list of available bed in particular room (No have any available bed)
Using: R016



```
SQL> DECLARE
  2  totalbed number(8);
  3  query VARCHAR(6);
  4  BEGIN
  5  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  6  DBMS_OUTPUT.PUT_LINE('Available bed query');
  7  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
  8  query:='&roomid_or_ALL';
  9  IF UPPER(query)='ALL' THEN
 10  totalbed:=total_bed_available(query);
 11  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
 12  DBMS_OUTPUT.PUT_LINE('All bed in hospital available are total of '||TO_NUMBER(totalbed,'9999'));
 13  ELSIF REGEXP_LIKE(query,'^R\d{3}$') THEN
 14  totalbed:=total_bed_available(query);
 15  DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
 16  DBMS_OUTPUT.PUT_LINE('All bed in room with ID '||query||' have total of '||TO_NUMBER(totalbed,'9999'));
 17  ELSE
 18  DBMS_OUTPUT.PUT_LINE('Invalid input. Key in again');
 19  END IF;
 20  END;
 21  /
Enter value for roomid_or_all: R016
old   8: query:='&roomid_or_ALL';
new   8: query:='R016';
--------------------------------------------------------------------------------
Available bed query
--------------------------------------------------------------------------------
Room                |    Bed Available
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
All bed in room with ID R016 have total of 0
```

Invalid input
Using: d



```
Enter value for roomid_or_all: d
old   8: query:='&roomid_or_ALL';
new   8: query:='d';
--------------------------------------------------------------------------------
Available bed query
--------------------------------------------------------------------------------
Invalid input. Key in again

PL/SQL procedure successfully completed.
```

F2

Calculate the total staff payment

- This function provides to hospital financial department and management to estimate the total staff payment.
- This function can use to analyse the financial output to staff.
- The function accepts three parameter, categories, duration, and epf percentage.
  For example, 'edn' stands for calculate all staff payment (the value inside the parameter is interchangeable, 'den' will give the same output as well), 'ed' stand for calculate admin staff and doctor, 'n' stands for inly calculate for nurse.
  For example, duration=2 stand for calculate 2-month, duration=12 stand for calculate one year payment
  For example, epf=11 mean 11% epf given to the staff.

Function

```
--Function
CREATE OR REPLACE FUNCTION total_staff_payment
(
   categories VARCHAR, --<e admin staff, n nurse, d doctor>
   duration Integer, --<1 -1month, 2 -2months, etc>
   epf Number  --Percentage
)
RETURN Number
IS
total Number;
totale Number;
totaln Number;
totald Number;
CURSOR nptr is
        SELECT employee_id, salary FROM employee, nurse WHERE employee_id=nurse_id AND leave_date is null;
CURSOR dptr is
        SELECT employee_id, salary FROM employee, doctor WHERE employee_id=doctor_id AND leave_date is null;
CURSOR eptr is
        SELECT employee_id, salary FROM employee WHERE leave_date is null MINUS
        SELECT employee_id, salary FROM employee, doctor WHERE employee_id=doctor_id AND leave_date is null
MINUS
        SELECT employee_id, salary FROM employee, nurse WHERE employee_id=nurse_id AND leave_date is null;
BEGIN
   total:=0;
```

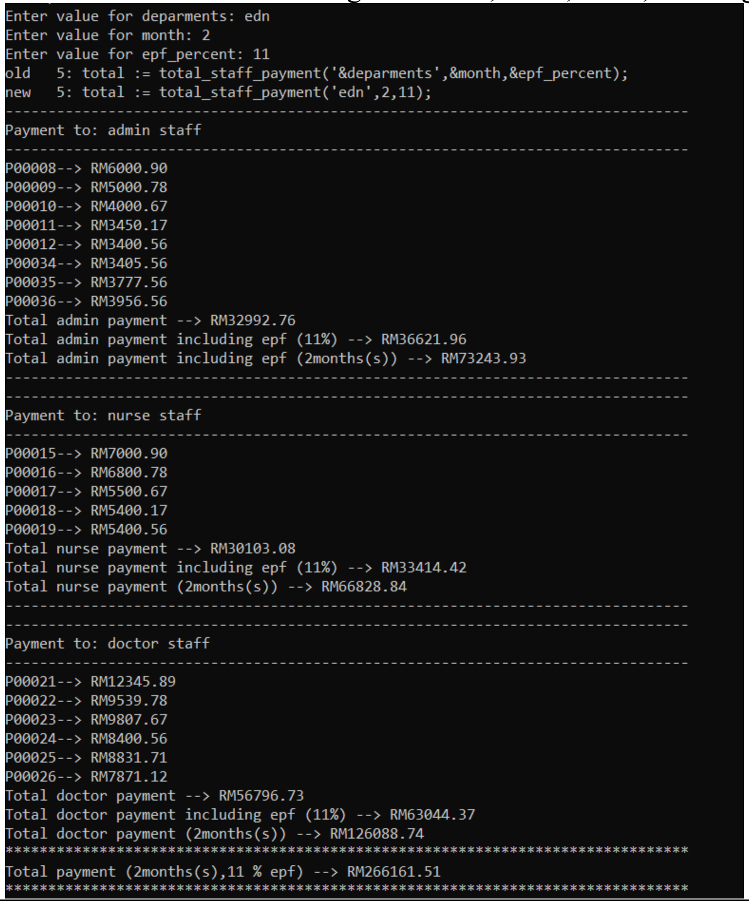
```
    totale:=0;
    totaln:=0;
    totald:=0;
    IF INSTR(categories,'e')!=0 THEN
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('Payment to: admin staff');
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        FOR e IN eptr
        LOOP
            DBMS_OUTPUT.PUT_LINE(e.employee_id||'--> RM'||TRIM(TO_CHAR(e.salary,'999999999.99')));
            totale:=totale+e.salary;
        END LOOP;
        DBMS_OUTPUT.PUT_LINE('Total admin payment --> RM'||TRIM(TO_CHAR(totale,'999999999.99')));
        totale:=totale+totale*epf/100;
        DBMS_OUTPUT.PUT_LINE('Total admin payment including epf ('||epf||'%) -->
RM'||TRIM(TO_CHAR(totale,'999999999.99')));
        totale:=totale*duration;
        DBMS_OUTPUT.PUT_LINE('Total admin payment including epf ('||duration||'months(s)) -->
RM'||TRIM(TO_CHAR(totale,'999999999.99')));
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    END IF;

    IF INSTR(categories,'n')!=0 THEN
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('Payment to: nurse staff');
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        FOR n IN nptr
        LOOP
            DBMS_OUTPUT.PUT_LINE(n.employee_id||'--> RM'||TRIM(TO_CHAR(n.salary,'999999999.99')));
            totaln:=totaln+n.salary;
        END LOOP;
        DBMS_OUTPUT.PUT_LINE('Total nurse payment --> RM'||TRIM(TO_CHAR(totaln,'999999999.99')));
        totaln:=totaln+totaln*epf/100;
        DBMS_OUTPUT.PUT_LINE('Total nurse payment including epf ('||epf||'%) -->
RM'||TRIM(TO_CHAR(totaln,'999999999.99')));
```

```
        totaln:=totaln*duration;
        DBMS_OUTPUT.PUT_LINE('Total nurse payment ('||duration||'months(s)) -->
RM'||TRIM(TO_CHAR(totaln,'999999999.99')));
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
    END IF;

    IF INSTR(categories,'d')!=0 THEN
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        DBMS_OUTPUT.PUT_LINE('Payment to: doctor staff');
        DBMS_OUTPUT.PUT_LINE(rpad('-',80,'-'));
        FOR d IN dptr
        LOOP
            DBMS_OUTPUT.PUT_LINE(d.employee_id||'--> RM'||TRIM(TO_CHAR(d.salary,'999999999.99')));
            totald:=totald+d.salary;
        END LOOP;
        DBMS_OUTPUT.PUT_LINE('Total doctor payment --> RM'||TRIM(TO_CHAR(totald,'999999999.99')));
        totald := totald+totald*epf/100;
        DBMS_OUTPUT.PUT_LINE('Total doctor payment including epf ('||epf||'%) -->
RM'||TRIM(TO_CHAR(totald,'999999999.99')));
        totald := totald*duration;
        DBMS_OUTPUT.PUT_LINE('Total doctor payment ('||duration||'months(s)) -->
RM'||TRIM(TO_CHAR(totald,'999999999.99')));
    END IF;
    total:=totale+totaln+totald;
  RETURN total;
END;
/
--Execution
DECLARE
    total number(10,2);
BEGIN
    total := total_staff_payment('edn',2,11);

    DBMS_OUTPUT.PUT_LINE(rpad('*',80,'*'));
    DBMS_OUTPUT.PUT_LINE('Total payment (2months(s),11 % epf) -->
RM'||TRIM(TO_CHAR(total,'999999999.99')));
    DBMS_OUTPUT.PUT_LINE(rpad('*',80,'*'));
END;
/
```

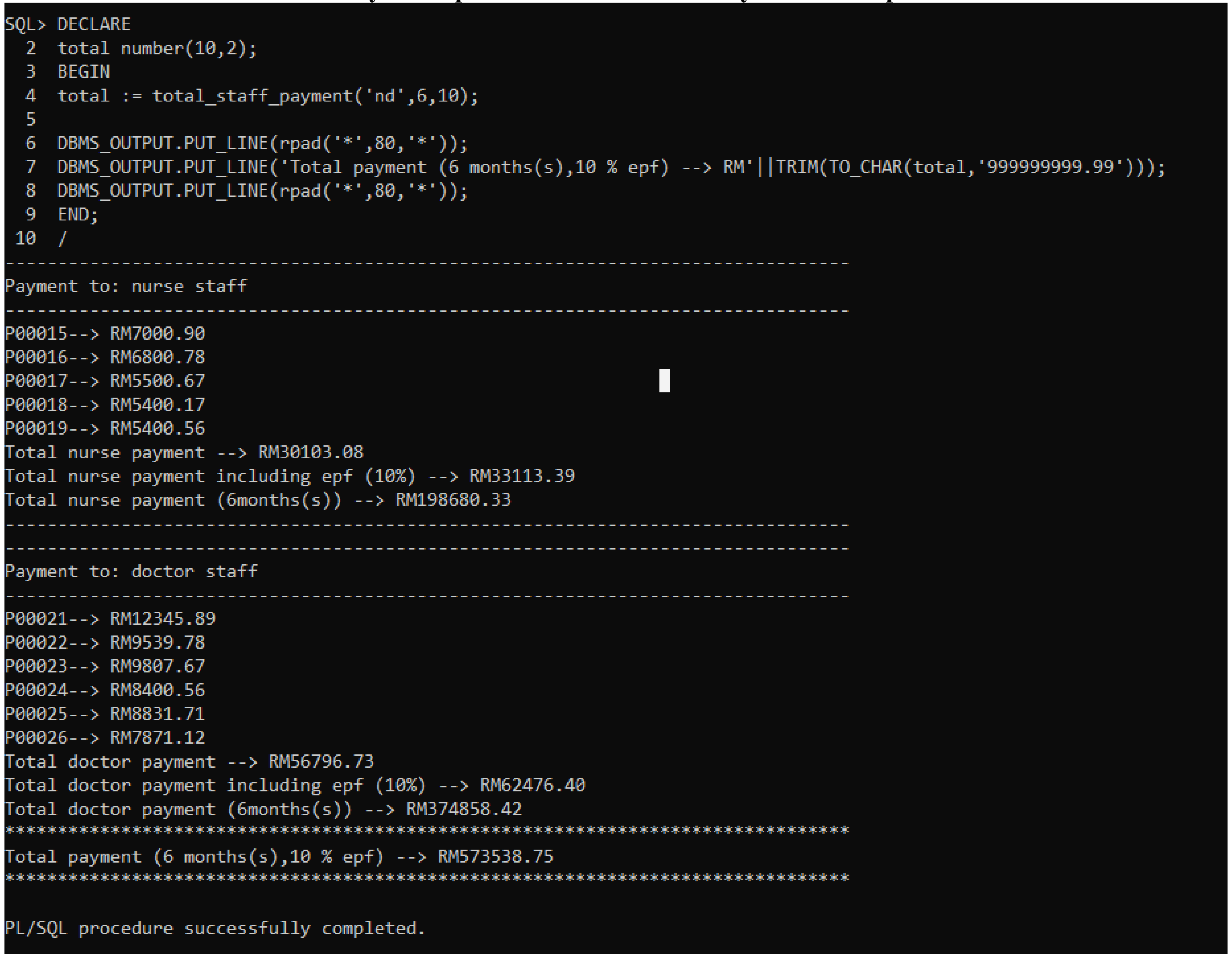
Screenshot

Calculate all staff salary in hospital with 2 months salary and 11 % epf.

Parameter :‘edn’ can be changed to ‘end’, ‘den’, ‘ned’,’nde’ regardless of arrangement

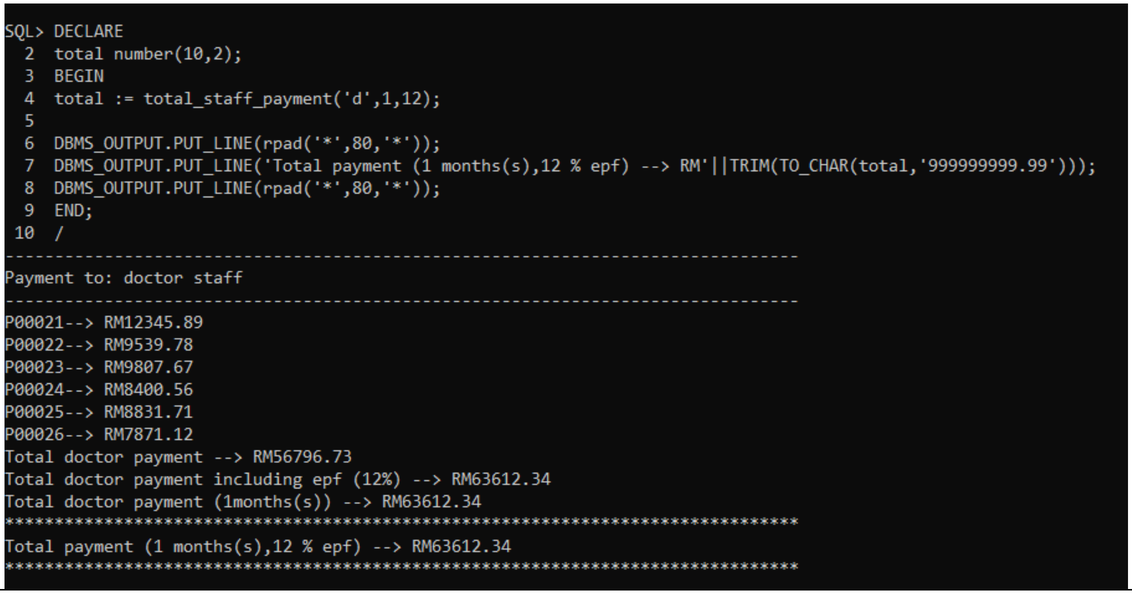


```
Enter value for deparments: edn
Enter value for month: 2
Enter value for epf_percent: 11
old   5: total := total_staff_payment('&deparments',&month,&epf_percent);
new   5: total := total_staff_payment('edn',2,11);
--------------------------------------------------------------------------------
Payment to: admin staff
--------------------------------------------------------------------------------
P00008--> RM6000.90
P00009--> RM5000.78
P00010--> RM4000.67
P00011--> RM3450.17
P00012--> RM3400.56
P00034--> RM3405.56
P00035--> RM3777.56
P00036--> RM3956.56
Total admin payment --> RM32992.76
Total admin payment including epf (11%) --> RM36621.96
Total admin payment including epf (2months(s)) --> RM73243.93
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Payment to: nurse staff
--------------------------------------------------------------------------------
P00015--> RM7000.90
P00016--> RM6800.78
P00017--> RM5500.67
P00018--> RM5400.17
P00019--> RM5400.56
Total nurse payment --> RM30103.08
Total nurse payment including epf (11%) --> RM33414.42
Total nurse payment (2months(s)) --> RM66828.84
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Payment to: doctor staff
--------------------------------------------------------------------------------
P00021--> RM12345.89
P00022--> RM9539.78
P00023--> RM9807.67
P00024--> RM8400.56
P00025--> RM8831.71
P00026--> RM7871.12
Total doctor payment --> RM56796.73
Total doctor payment including epf (11%) --> RM63044.37
Total doctor payment (2months(s)) --> RM126088.74
********************************************************************************
Total payment (2months(s),11 % epf) --> RM266161.51
********************************************************************************
```

	Calculate nurse and doctor salary in hospital with 6 months' salary and 10 % epf 	

```
SQL> DECLARE
  2  total number(10,2);
  3  BEGIN
  4  total := total_staff_payment('nd',6,10);
  5
  6  DBMS_OUTPUT.PUT_LINE(rpad('*',80,'*'));
  7  DBMS_OUTPUT.PUT_LINE('Total payment (6 months(s),10 % epf) --> RM'||TRIM(TO_CHAR(total,'999999999.99')));
  8  DBMS_OUTPUT.PUT_LINE(rpad('*',80,'*'));
  9  END;
 10  /
--------------------------------------------------------------------------------
Payment to: nurse staff
--------------------------------------------------------------------------------
P00015--> RM7000.90
P00016--> RM6800.78
P00017--> RM5500.67
P00018--> RM5400.17
P00019--> RM5400.56
Total nurse payment --> RM30103.08
Total nurse payment including epf (10%) --> RM33113.39
Total nurse payment (6months(s)) --> RM198680.33
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
Payment to: doctor staff
--------------------------------------------------------------------------------
P00021--> RM12345.89
P00022--> RM9539.78
P00023--> RM9807.67
P00024--> RM8400.56
P00025--> RM8831.71
P00026--> RM7871.12
Total doctor payment --> RM56796.73
Total doctor payment including epf (10%) --> RM62476.40
Total doctor payment (6months(s)) --> RM374858.42
********************************************************************************
Total payment (6 months(s),10 % epf) --> RM573538.75
********************************************************************************

PL/SQL procedure successfully completed.
```

Calculate only doctor salary in hospital with 1 month' salary and 12 % epf



```
SQL> DECLARE
total number(10,2);
BEGIN
total := total_staff_payment('d',1,12);

DBMS_OUTPUT.PUT_LINE(rpad('*',80,'*'));
DBMS_OUTPUT.PUT_LINE('Total payment (1 months(s),12 % epf) --> RM'||TRIM(TO_CHAR(total,'999999999.99')));
DBMS_OUTPUT.PUT_LINE(rpad('*',80,'*'));
END;
/
--------------------------------------------------------------------------------
Payment to: doctor staff
--------------------------------------------------------------------------------
P00021--> RM12345.89
P00022--> RM9539.78
P00023--> RM9807.67
P00024--> RM8400.56
P00025--> RM8831.71
P00026--> RM7871.12
Total doctor payment --> RM56796.73
Total doctor payment including epf (12%) --> RM63612.34
Total doctor payment (1months(s)) --> RM63612.34
********************************************************************************
Total payment (1 months(s),12 % epf) --> RM63612.34
********************************************************************************
```

F3

<u>Calculate the most preferable doctor-department by public</u>

- This function can calculate the most preferable department by public in the hospital.
- This function assist hospital management to know which department in their hospital which public more likely come for and can make more staff arrangement in that department, and buy more related medicine equipment to get ready.
- This function and output the result for reward encourage the doctors in that department to provide more excellent services.
- This involves a nested call of function. The department_count function will return the count in service record(the department of a involved doctor). The most_preferable_department will return the result to where it executed.
- When there are two department with same result. Both of them will be show together.

<u>Function</u>

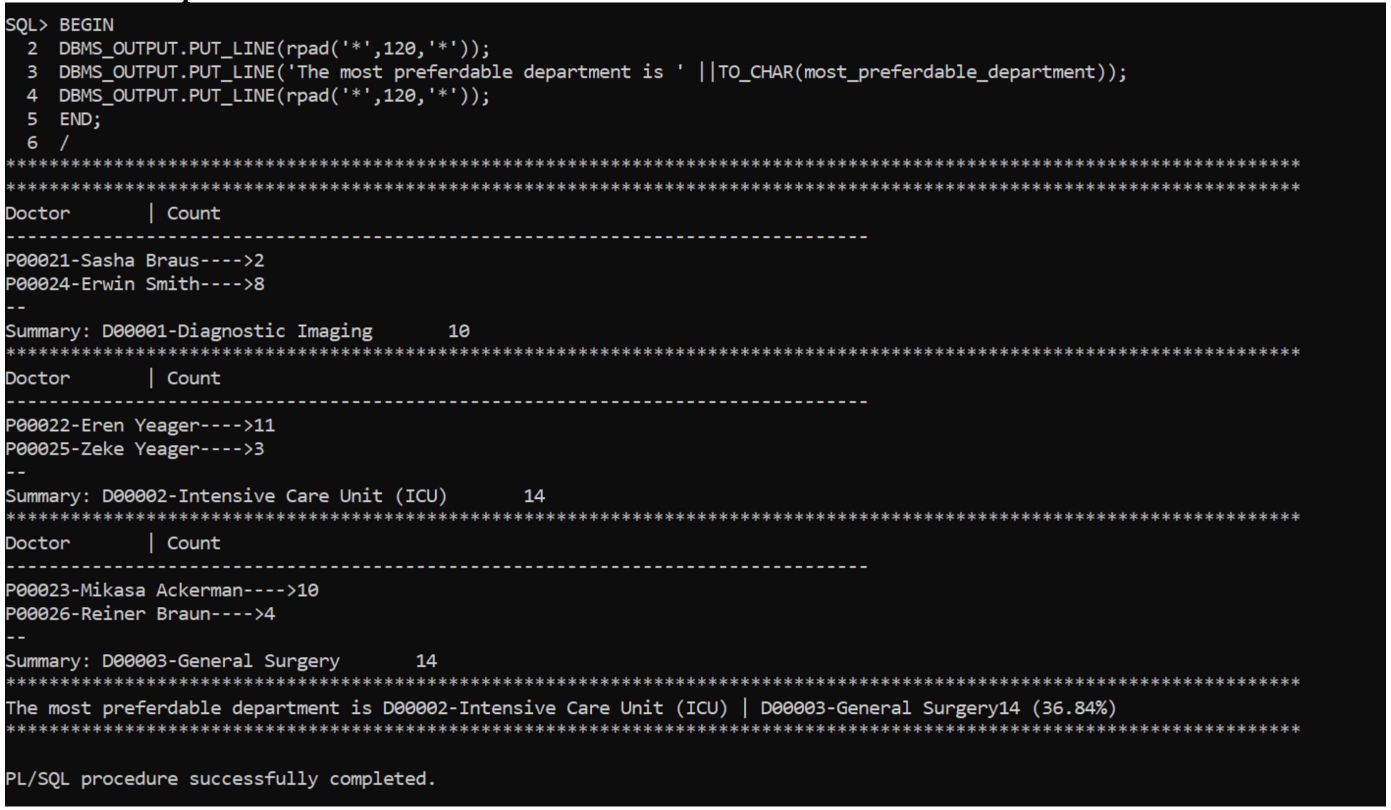
```
--Function - count each department have the high number of services provided
CREATE OR REPLACE FUNCTION most_preferable_department
RETURN VARCHAR
IS
```

```
temp Number;
compare Number;
text VARCHAR(1000);
total Number;
CURSOR pointers is
    SELECT DISTINCT dt.department_id AS deptid, dt.name AS dname
    FROM doctor d, employee e, department dt
    WHERE d.doctor_id=e.employee_id
    AND e.department_id=dt.department_id
    ORDER BY 1 ASC;
BEGIN
    text:='';
    compare:=0;
    total:=0;
    DBMS_OUTPUT.PUT_LINE(rpad('*',120,'*'));
    FOR ptr IN pointers
    LOOP
    temp:=department_count(ptr.deptid);
    total:=total+temp;
    DBMS_OUTPUT.PUT_LINE('--');
    DBMS_OUTPUT.PUT_LINE('Summary: '||ptr.deptid||'-'||ptr.dname||'       '||TO_CHAR(temp));
    IF compare<temp THEN
        compare:=temp;
        text:=CONCAT(CONCAT(ptr.deptid,'-'),ptr.dname);
    ELSIF compare=temp THEN
        text:= CONCAT(text,CONCAT(' | ',CONCAT(CONCAT(ptr.deptid,'-'),ptr.dname)));
    END IF;
    DBMS_OUTPUT.PUT_LINE(rpad('*',120,'*'));
    END LOOP;
    text:= CONCAT(CONCAT(text,TO_CHAR(compare)), CONCAT(CONCAT('
(',TRIM(TO_CHAR(compare/total*100,'99999.99'))),'%)'));
    return text;
END;
/
```

```
--Execution
BEGIN
    DBMS_OUTPUT.PUT_LINE(rpad('*',120,'*'));
    DBMS_OUTPUT.PUT_LINE('The most preferable department is ' ||TO_CHAR(most_preferable_department));
    DBMS_OUTPUT.PUT_LINE(rpad('*',120,'*'));
END;
/
```

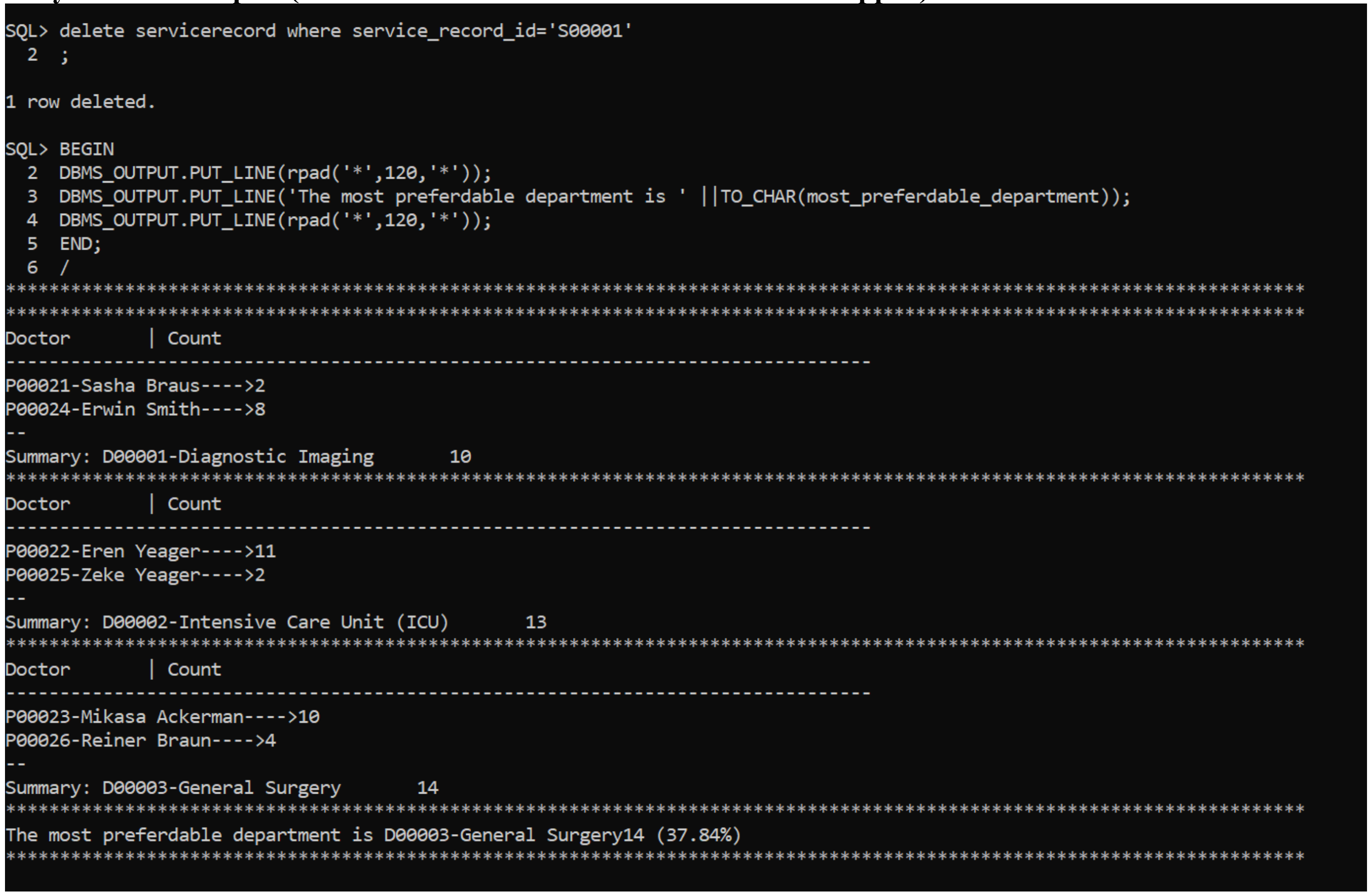
<u>Screenshot</u>

When two output same



```
SQL> BEGIN
DBMS_OUTPUT.PUT_LINE(rpad('*',120,'*'));
DBMS_OUTPUT.PUT_LINE('The most preferdable department is ' ||TO_CHAR(most_preferdable_department));
DBMS_OUTPUT.PUT_LINE(rpad('*',120,'*'));
END;
/
************************************************************************************************************************
************************************************************************************************************************
Doctor       | Count
--------------------------------------------------------------------------------
P00021-Sasha Braus---->2
P00024-Erwin Smith---->8
--
Summary: D00001-Diagnostic Imaging       10
************************************************************************************************************************
Doctor       | Count
--------------------------------------------------------------------------------
P00022-Eren Yeager---->11
P00025-Zeke Yeager---->3
--
Summary: D00002-Intensive Care Unit (ICU)       14
************************************************************************************************************************
Doctor       | Count
--------------------------------------------------------------------------------
P00023-Mikasa Ackerman---->10
P00026-Reiner Braun---->4
--
Summary: D00003-General Surgery       14
************************************************************************************************************************
The most preferdable department is D00002-Intensive Care Unit (ICU) | D00003-General Surgery14 (36.84%)
************************************************************************************************************************

PL/SQL procedure successfully completed.
```

Only have one output. (If a service record of doctor in D00002 is dropped)



```
SQL> delete servicerecord where service_record_id='S00001'
  2  ;

1 row deleted.

SQL> BEGIN
  2  DBMS_OUTPUT.PUT_LINE(rpad('*',120,'*'));
  3  DBMS_OUTPUT.PUT_LINE('The most preferdable department is ' ||TO_CHAR(most_preferdable_department));
  4  DBMS_OUTPUT.PUT_LINE(rpad('*',120,'*'));
  5  END;
  6  /
************************************************************************************************************************
************************************************************************************************************************
Doctor      | Count
--------------------------------------------------------------------------------
P00021-Sasha Braus---->2
P00024-Erwin Smith---->8
--
Summary: D00001-Diagnostic Imaging      10
************************************************************************************************************************
Doctor      | Count
--------------------------------------------------------------------------------
P00022-Eren Yeager---->11
P00025-Zeke Yeager---->2
--
Summary: D00002-Intensive Care Unit (ICU)      13
************************************************************************************************************************
Doctor      | Count
--------------------------------------------------------------------------------
P00023-Mikasa Ackerman---->10
P00026-Reiner Braun---->4
--
Summary: D00003-General Surgery      14
************************************************************************************************************************
The most preferdable department is D00003-General Surgery14 (37.84%)
************************************************************************************************************************
```

F4

Calculate and analyze diseases among different age interval function

- This function can analyze the disease among the different age interval.
- This can help hospital to get ready of related specialist and medical equipment as well based on the suitability on the age.
- This function can give hospital to publish advertisement to increase public concern about particular diseases.
- The function of disease_count will return particular disease appear in service record in an age interval. A For loop is needed for iteration the age interval and call the disease_count function.

Function

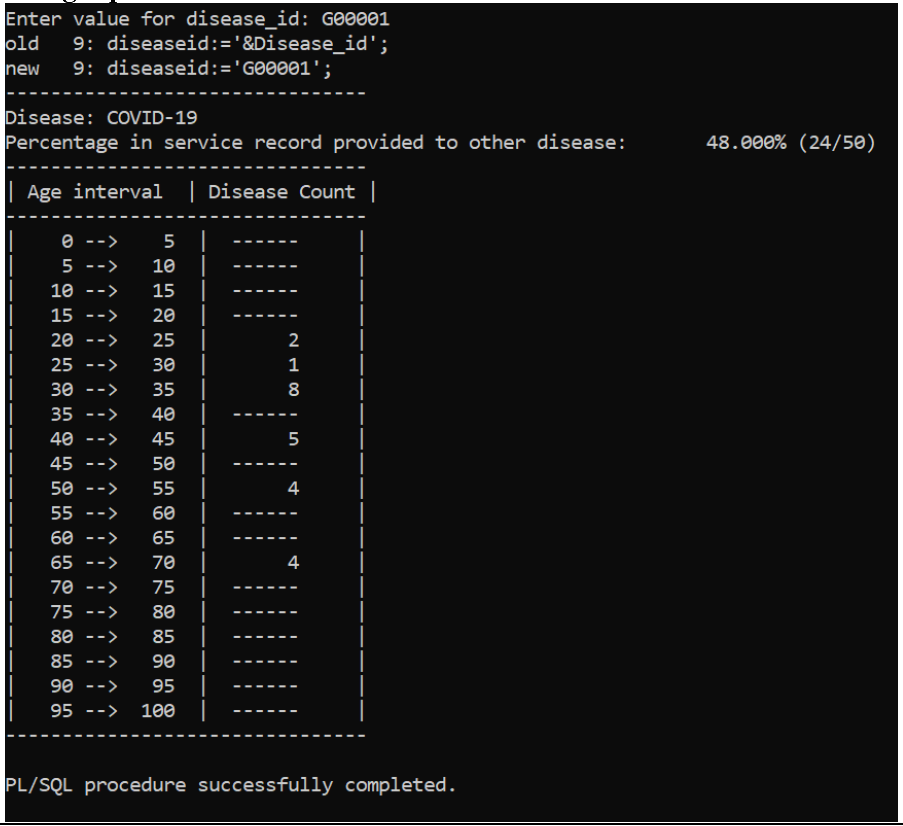
```
--Function -Return disease number in record
CREATE OR REPLACE FUNCTION disease_count
(
    diseaseid VARCHAR,
    interval_start Number,
    interval_end Number
)
RETURN NUMBER
IS
    counts Number;
    CURSOR pointers is
        SELECT g.disease_id ||'-'|| g.name AS Diseases, TRUNC((SYSDATE-p.birth_date)/365.25)  AS AGE,
COUNT(*) AS times
        FROM disease g, servicerecord s, patient pt, person p, admission a
        WHERE g.disease_id = s.disease_id
        AND s.admission_id = a.admission_id
        AND a.patient_id = pt.patient_id
        AND pt.patient_id = p.person_id
        AND g.disease_id = diseaseid
        GROUP BY g.name,p.birth_date,g.disease_id, a.admission_id
        ORDER BY g.disease_id,age;
BEGIN
    counts:=0;
    FOR ptr IN pointers
    LOOP
        IF ptr.age >= interval_start AND ptr.age<interval_end THEN
```

```
            counts := counts+ptr.times;
        END IF;
    END LOOP;
    return counts;
END;
/

--Execution with for loop
DECLARE
    a Number;
    diseaseid VARCHAR(6);
    diseasename VARCHAR(20);
    target Number;
    total Number;
BEGIN
    a := 0;
    diseaseid:='&Disease_id';
    SELECT d.name into diseasename FROM disease d WHERE d.disease_id=diseaseid;


    SELECT COUNT(*) into total
    FROM servicerecord s
    WHERE s.disease_id is not null;

    SELECT COUNT(*) into target
    FROM servicerecord s
    WHERE s.disease_id = diseaseid;
    DBMS_OUTPUT.PUT_LINE(rpad('-',32,'-'));
    DBMS_OUTPUT.PUT_LINE('Disease: '|| diseasename);
    DBMS_OUTPUT.PUT_LINE('Percentage in service record provided to other disease: '||
TO_CHAR(target/total*100,'9999999.990')||'% ('||target||'/'||total||')');
    DBMS_OUTPUT.PUT_LINE(rpad('-',32,'-'));
    DBMS_OUTPUT.PUT_LINE('| Age interval  | Disease Count |');
    DBMS_OUTPUT.PUT_LINE(rpad('-',32,'-'));
    WHILE a < 100 LOOP
```

```
        DBMS_OUTPUT.PUT_LINE('| '||TO_CHAR(a,'999')||' --> '||TO_CHAR(a+5,'999')||'  |  '||
        CASE WHEN TRIM(TO_CHAR(disease_count(diseaseid,a,a+5),'99999'))='0'THEN '------' ELSE
TO_CHAR(disease_count(diseaseid,a,a+5),'99999')END||'     |');
        a := a+5;
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(rpad('-',32,'-'));
END;
/
```

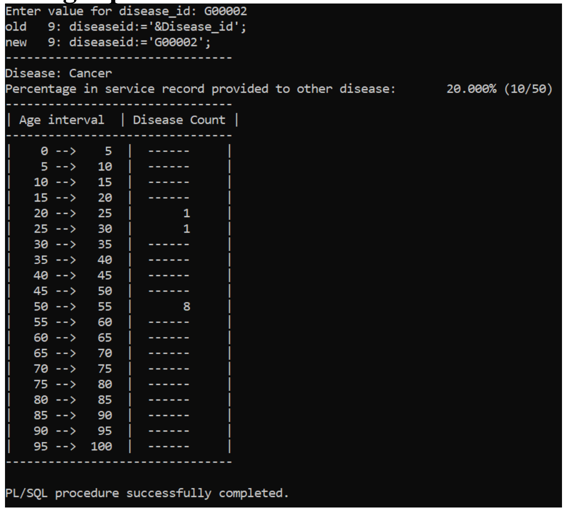
Screenshot

Using input: G00001



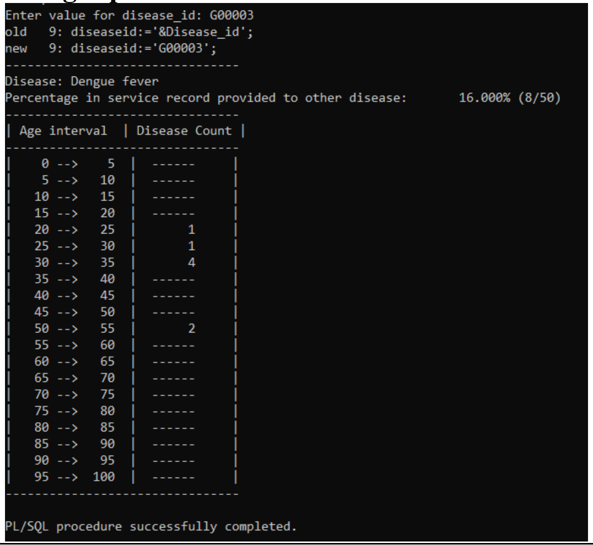
```
Enter value for disease_id: G00001
old   9: diseaseid:='&Disease_id';
new   9: diseaseid:='G00001';
--------------------------------
Disease: COVID-19
Percentage in service record provided to other disease:       48.000% (24/50)
--------------------------------
| Age interval  | Disease Count |
--------------------------------
|    0 -->    5  |  ------     |
|    5 -->   10  |  ------     |
|   10 -->   15  |  ------     |
|   15 -->   20  |  ------     |
|   20 -->   25  |       2     |
|   25 -->   30  |       1     |
|   30 -->   35  |       8     |
|   35 -->   40  |  ------     |
|   40 -->   45  |       5     |
|   45 -->   50  |  ------     |
|   50 -->   55  |       4     |
|   55 -->   60  |  ------     |
|   60 -->   65  |  ------     |
|   65 -->   70  |       4     |
|   70 -->   75  |  ------     |
|   75 -->   80  |  ------     |
|   80 -->   85  |  ------     |
|   85 -->   90  |  ------     |
|   90 -->   95  |  ------     |
|   95 -->  100  |  ------     |
--------------------------------

PL/SQL procedure successfully completed.
```

Using input: G00002



```
Enter value for disease_id: G00002
old   9: diseaseid:='&Disease_id';
new   9: diseaseid:='G00002';
--------------------------------
Disease: Cancer
Percentage in service record provided to other disease:       20.000% (10/50)
--------------------------------
| Age interval  | Disease Count |
--------------------------------
|    0 -->    5  |  ------     |
|    5 -->   10  |  ------     |
|   10 -->   15  |  ------     |
|   15 -->   20  |  ------     |
|   20 -->   25  |       1     |
|   25 -->   30  |       1     |
|   30 -->   35  |  ------     |
|   35 -->   40  |  ------     |
|   40 -->   45  |  ------     |
|   45 -->   50  |  ------     |
|   50 -->   55  |       8     |
|   55 -->   60  |  ------     |
|   60 -->   65  |  ------     |
|   65 -->   70  |  ------     |
|   70 -->   75  |  ------     |
|   75 -->   80  |  ------     |
|   80 -->   85  |  ------     |
|   85 -->   90  |  ------     |
|   90 -->   95  |  ------     |
|   95 -->  100  |  ------     |
--------------------------------

PL/SQL procedure successfully completed.
```

Using input: G00003

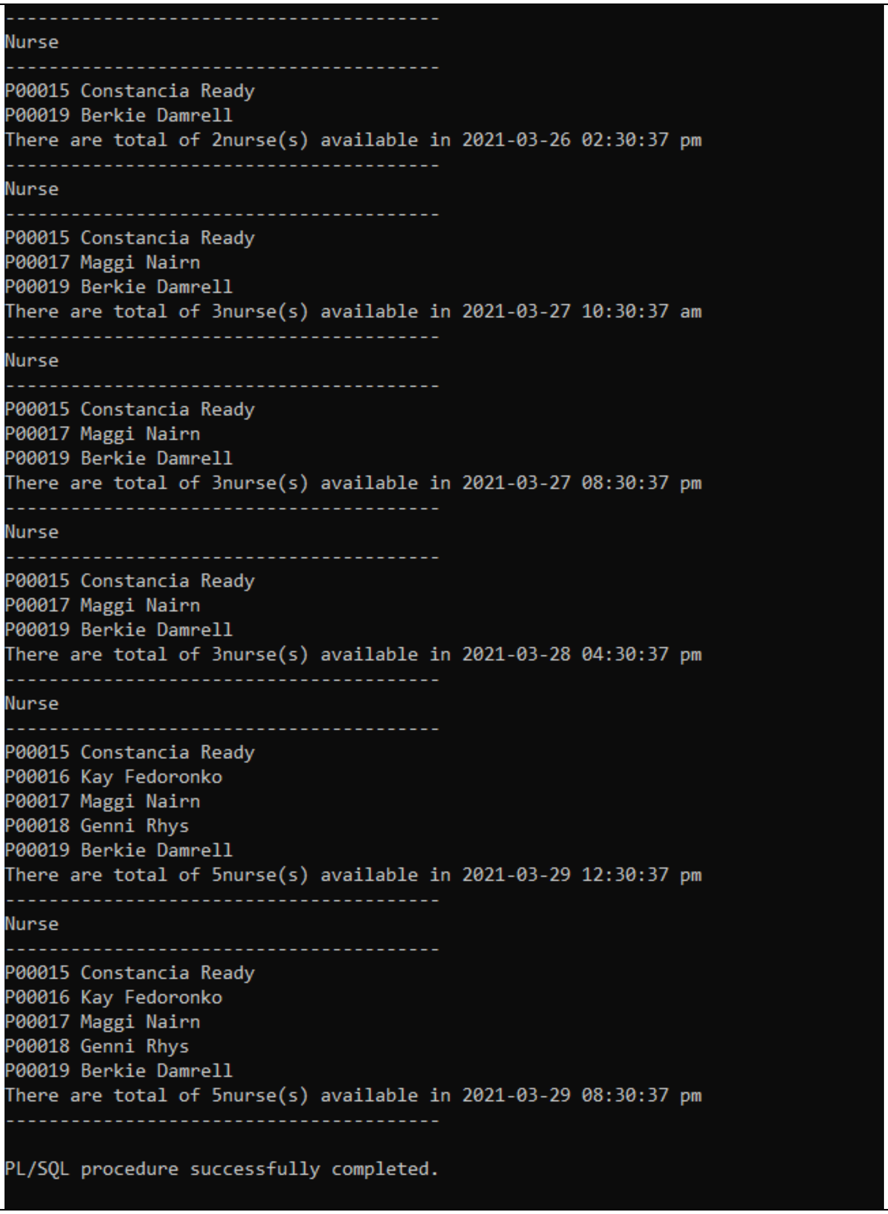


```
Enter value for disease_id: G00003
old   9: diseaseid:='&Disease_id';
new   9: diseaseid:='G00003';
--------------------------------
Disease: Dengue fever
Percentage in service record provided to other disease:       16.000% (8/50)
--------------------------------
| Age interval  | Disease Count |
--------------------------------
|    0 -->    5  |  ------     |
|    5 -->   10  |  ------     |
|   10 -->   15  |  ------     |
|   15 -->   20  |  ------     |
|   20 -->   25  |       1     |
|   25 -->   30  |       1     |
|   30 -->   35  |       4     |
|   35 -->   40  |  ------     |
|   40 -->   45  |  ------     |
|   45 -->   50  |  ------     |
|   50 -->   55  |       2     |
|   55 -->   60  |  ------     |
|   60 -->   65  |  ------     |
|   65 -->   70  |  ------     |
|   70 -->   75  |  ------     |
|   75 -->   80  |  ------     |
|   80 -->   85  |  ------     |
|   85 -->   90  |  ------     |
|   90 -->   95  |  ------     |
|   95 -->  100  |  ------     |
--------------------------------

PL/SQL procedure successfully completed.
```

F5	**Count the nurse which not perform any service in a time. (Available nurse)**
	• This is helping the hospital to know which time will lacking of nurse.
	• This function needs to have a time input and the function will search of the available nurse in the time and return the total of nurse count, meanwhile, it will list out the related available nurse.
	• Human resources of hospital can try to recruit new nurse when there always not sufficient number of nurse.
	• A for loop is needed to loop this function to get a list of nurses available for different time.
	• An array can define to easy sort out the timeslot easily.
	
	Function
	--Function
	CREATE OR REPLACE FUNCTION nurse_count
	(
	    ctime TIMESTAMP
	)
	RETURN NUMBER
	IS
	    counts Number;
	    CURSOR pointers is
	        SELECT p.person_id AS pid, p.first_name AS pfn,p.last_name AS pln
	        FROM person p,nurse n, employee e
	        WHERE n.nurse_id=e.employee_id
	        AND e.employee_id=p.person_id
	        AND e.leave_date IS NULL
	        MINUS
	        SELECT p.person_id, p.first_name,p.last_name
	        FROM nurse n, employee e, person p, servicerecord s
	        WHERE s.nurse_id = n.nurse_id
	        AND n.nurse_id=e.employee_id
	        AND e.employee_id=p.person_id
	        AND e.leave_date IS NULL
	        AND ctime > s.start_time
	        AND ctime < CASE WHEN s.end_time is null then SYSTIMESTAMP+1 ELSE s.end_time END;
	BEGIN
	    counts:=0;	

```
    DBMS_OUTPUT.PUT_LINE('Nurse');
    DBMS_OUTPUT.PUT_LINE(rpad('-',40,'-'));
    FOR ptr IN pointers
    LOOP
        counts:=counts+1;
        DBMS_OUTPUT.PUT_LINE(ptr.pid||' '||ptr.pfn||' '||ptr.pln);
    END LOOP;
    return counts;
END;
/

--Execution - For loop with array
DECLARE
  type array_t is varray(6) of Number;
  array array_t := array_t(-30,-10,0,20,40,48);
BEGIN
    FOR i in 1..array.count LOOP
        DBMS_OUTPUT.PUT_LINE(rpad('-',40,'-'));
        DBMS_OUTPUT.PUT_LINE('There are total of '||nurse_count(SYSTIMESTAMP +
TO_NUMBER(array(i)/24))||' nurse(s) available in '||TO_CHAR(SYSTIMESTAMP +
TO_NUMBER(array(i)/24),'YYYY-MM-DD HH:MI:SS pm'));
    END LOOP;
    DBMS_OUTPUT.PUT_LINE(rpad('-',40,'-'));
END;
/
```

Screenshot
The for loop is use to show the nurse available in certain time.
The execution can be also change to looping every 30 minutes to get the list of schedules too.
The array is declared and be use.
The output below shows the nurse available 30 hours ago, 10 hours ago, current time, 20 hour later, 40 hours later and 48 hours later.

```
----------------------------------------
Nurse
----------------------------------------
P00015 Constancia Ready
P00019 Berkie Damrell
There are total of 2nurse(s) available in 2021-03-26 02:30:37 pm
----------------------------------------
Nurse
----------------------------------------
P00015 Constancia Ready
P00017 Maggi Nairn
P00019 Berkie Damrell
There are total of 3nurse(s) available in 2021-03-27 10:30:37 am
----------------------------------------
Nurse
----------------------------------------
P00015 Constancia Ready
P00017 Maggi Nairn
P00019 Berkie Damrell
There are total of 3nurse(s) available in 2021-03-27 08:30:37 pm
----------------------------------------
Nurse
----------------------------------------
P00015 Constancia Ready
P00017 Maggi Nairn
P00019 Berkie Damrell
There are total of 3nurse(s) available in 2021-03-28 04:30:37 pm
----------------------------------------
Nurse
----------------------------------------
P00015 Constancia Ready
P00016 Kay Fedoronko
P00017 Maggi Nairn
P00018 Genni Rhys
P00019 Berkie Damrell
There are total of 5nurse(s) available in 2021-03-29 12:30:37 pm
----------------------------------------
Nurse
----------------------------------------
P00015 Constancia Ready
P00016 Kay Fedoronko
P00017 Maggi Nairn
P00018 Genni Rhys
P00019 Berkie Damrell
There are total of 5nurse(s) available in 2021-03-29 08:30:37 pm
----------------------------------------

PL/SQL procedure successfully completed.
```

Assignment Marking Scheme		
PART 1: (Group Assessment - 50%)		**Marks**
1.	**Scope of Work (5 marks)** Analyse requirements study (briefly explain the requirements/ office / business rules in the system). PLEASE INCLUDE ANY ASSUMPTIONS THAT YOU MAKE.	
2.	**ER model (10 marks)** You are required to design an ER diagram for the case study given, identify entities, identify relationships, identify associate attribute and determine keys. Check your ERD with the transaction requirements stated in the case.	
3.	**Redesign and EER (10 marks)** Redesign your ER diagram with the new requirements and extending the ERD to EER model, if any.	
4.	**Data Dictionary (10 marks)** Based on EER diagram that you created in part 4, create a data dictionary for the solution. (Make sure the data types (Oracle) selected are appropriate)	
5.	**Tables and records (5 marks)** Create all relations in ERD and insert the necessary records (Minimum 5 record for each table)	
6.	**Script (10 marks)** You are required to submit the SQL schema script with proper codes. Should include Integrity and referential integrity constraints. **Softcopy:** *Include the scripts in the submission*	
PART 1: Total Group Assessment - 50%		

PART 2: (Individual Assessment - 50%)
(Filled in all your group members name and ID)

Student Name	**1. Tan Jing Jie**	**2.**	**3.**	**4.**	
Student ID	**18ACB04560**				
Queries					

(30 marks)					
Stored Procedure (10 marks)					
Function (10 marks)					
PART 2: Total Individual Assessment – 50 marks					
PART 1 + PART 2 = 100 marks					