Faculty of Computing and Information Technology (FOCS)

Academic Year 2020/2021

BACS3183

Advanced Database Management

Programme	:	RDS3
Tutorial Group	:	G2
Date Submitted to Tutor	:	01-09-2021

Team Members:

No	Student Name	Student ID	Signature
1.	Tan Yi Hong	20WMR08890	[illegible]
2.	Tan Teoh Xin Ee	20WMR08887	[illegible]
3.	Tan Wei Siong	20WMR08888	Siong
4.	Nigel Lee Jian Hsee	20WMR08882	[illegible]

Declaration

We confirm that we have read and shall comply with all the terms and conditions of TAR University College's plagiarism policy.

We declare that this assignment is free from all forms of plagiarism and for all intents and purposes is my own properly derived work.

Signature :	[illegible]	[illegible]	Siong	[illegible]
Name :	Tan Yi Hong	Tan Teoh Xin Ee	Tan Wei Siong	Nigel Lee Jian Hsee
Date :	01-09-2021	01-09-2021	01-09-2021	01-09-2021

Table of Content

Chapter 1 Background of the System 4

Chapter 2 Entity-Relationship Modeling 5
2.1 ERD 5
2.2 Assumptions and Business Rules 5

Chapter 3 Data Definition 8

Chapter 4 Queries, Procedures, Triggers and Reports 12
4.1 (Tan Yi Hong) 12
4.1.1 Query 1: Top pet type that received treatment in each branch (Strategic) 12
4.1.2 Query 2: Appointment made on times of a day in each branch for last year (Tactical) 13
4.1.3 Query 3: Year 2020 first half sales vs second half sales in each branch (Operational) 15
4.1.4 Procedure 1: Add Appointment Record 16
4.1.5 Procedure 2: Delete Appointment Record 18
4.1.6 Trigger 1: Validate the proper date time on insertion of appointment 19
4.1.7 Trigger 2: Monitor the deletion of appointment record 20
4.1.8 Report 1: Summary report of Specific veterinarian with all of his/her transactions done in a year 21
4.1.9 Report 2: Detail report of Customer list in specific state with appointment made in a year 26
4.1.10 Report 3: On demand report of Customer list who do transaction for less than three times in a year 31

4.2 (Tan Teoh Xin Ee) 34
4.2.1 Query 1: Top Medicine Used in each branch (Strategic) 34
4.2.2 Query 2: Top Veterinarian in each branch (Tactical) 35
4.2.3 Query 3: Late Supplier List (Operational) 36
4.2.4 Procedure 1: Add Medical Supply 38
4.2.5 Procedure 2: Delete Medical Supply 39
4.2.6 Trigger 1: Add Stock Quantity (Add after purchasing stock) 40
4.2.7 Trigger 2: Minus Stock Quantity (Minus after adding transaction detail) 40
4.2.8 Report 1: Detail report of Veterinarian Performance 41
4.2.9 Report 2: Summary report of Supplier Profile 43
4.2.10 Report 3: On demand report of Inventory value 46

4.3 (Tan Wei Siong) 49
4.3.1 Query 1: Top and Least Treatment Last Year for Each Branch (Strategic) 49
4.3.2 Query 2: Selected Pet Type Age Group and Respective Visit Quantity(Tactical) 50
4.3.3 Query 3: Gross Profit and Net Profit Previous Year(Operational) 51
4.3.4 Procedure 1: Pet Registration 53
4.3.5 Procedure 2: Display Owner Information 54
4.3.6 Trigger 1: Check Owner Age 56
4.3.7 Trigger 2: Check Appointment Date Time 58
4.3.8 Report 1: Summary Report of Treatment Revenue in Selected Year 60
4.3.9 Report 2: Detail Report of Medic Performance 68
4.3.10 Report 3: On-demand Report of the Pet Treatment Detail 75

4.4 (Nigel Lee Jian Hsee) 80
4.2.1 Query 1: Top Three Pets Customer's Postcode for each Branch based on total transaction amount (Strategic) 80
4.4.2 Query 2: Rank top treatment, to medicine sales and total revenue of a branch 82
4.4.3 Query 3: View Customer Transaction History by entering Customer ID 85
4.4.4 Procedure 1: Create Transaction 89
4.4.5 Procedure 2: Add Transaction Detail 94
4.4.6 Procedure 3: Edit Transaction Detail 99
4.4.7 Trigger 1: Validate Payment Date Time 106
4.4.8 Trigger 2: Update medicine quantity and transaction total amount after editing transaction detail 107
4.4.9 Trigger 3: Update the stock and transaction total amount after deleting transaction detail 108
4.4.10 Trigger 4: Update total amount after inserting new transaction detail 108
4.4.11 Report 1: Summary report of Veterinarian Performance in a month 109
4.4.12 Report 2: On demand report of days that performed poorly for a specific with a date range 115
4.4.13 Report 3: Detail report of each Branch's Performance in a specific year 126

Chapter 5 Extra Effort Highlights 136

5.1 (Tan Yi Hong) 136
5.1.1 Views 136
5.1.2 User Defined Exception 138
5.1.3 Sequences 139
5.1.4 Triggers 140

5.2 (Tan Teoh Xin Ee)
5.2.1 Views 140
5.2.2 User Defined Exception 142
5.2.3 Sequence 142
5.2.4 Trigger 142

5.3 (Tan Wei Siong)
5.3.1 User Defined Exceptions 143
5.3.2 Sequence 143
5.3.3 Procedure 143
5.3.4 Trigger 145

5.4 (Nigel Lee Jian Hsee)
5.1.1 Views 145
5.1.2 User Defined Exceptions 146
5.4.3 Trigger 149

Chapter 1 Background of the System

The 4 Golden Duck Wellness Veterinary Clinic is a pet clinic operating from the year 2019 with a database of 4 main functions. 4 Golden Duck Wellness Veterinary Clinic has 3 three branches which are located in Kuala Lumpur, Penang, and Kedah.

Payment & Transaction

This function is to record every detail of a pet owner's transaction in the Wellness Veterinary Clinic. It can be used to track and monitor transactions of each pet owner. By recording all transactions, various reports can be generated to gain business insights. This function is able to calculate the total amount of a transaction. The transaction will include all the quantities of a medicine bought by the pet owner and treatment for their pets.

Appointment/Booking

The purpose of this function is to let customers make appointments for the treatment of their pets. This function can record all appointments made to make sure the service of treatment does not clash with each other and arrange the time of appointment schedule. Staff will be handling appointments and able to see the record in the tables to serve the customers accordingly.

Pet Registration

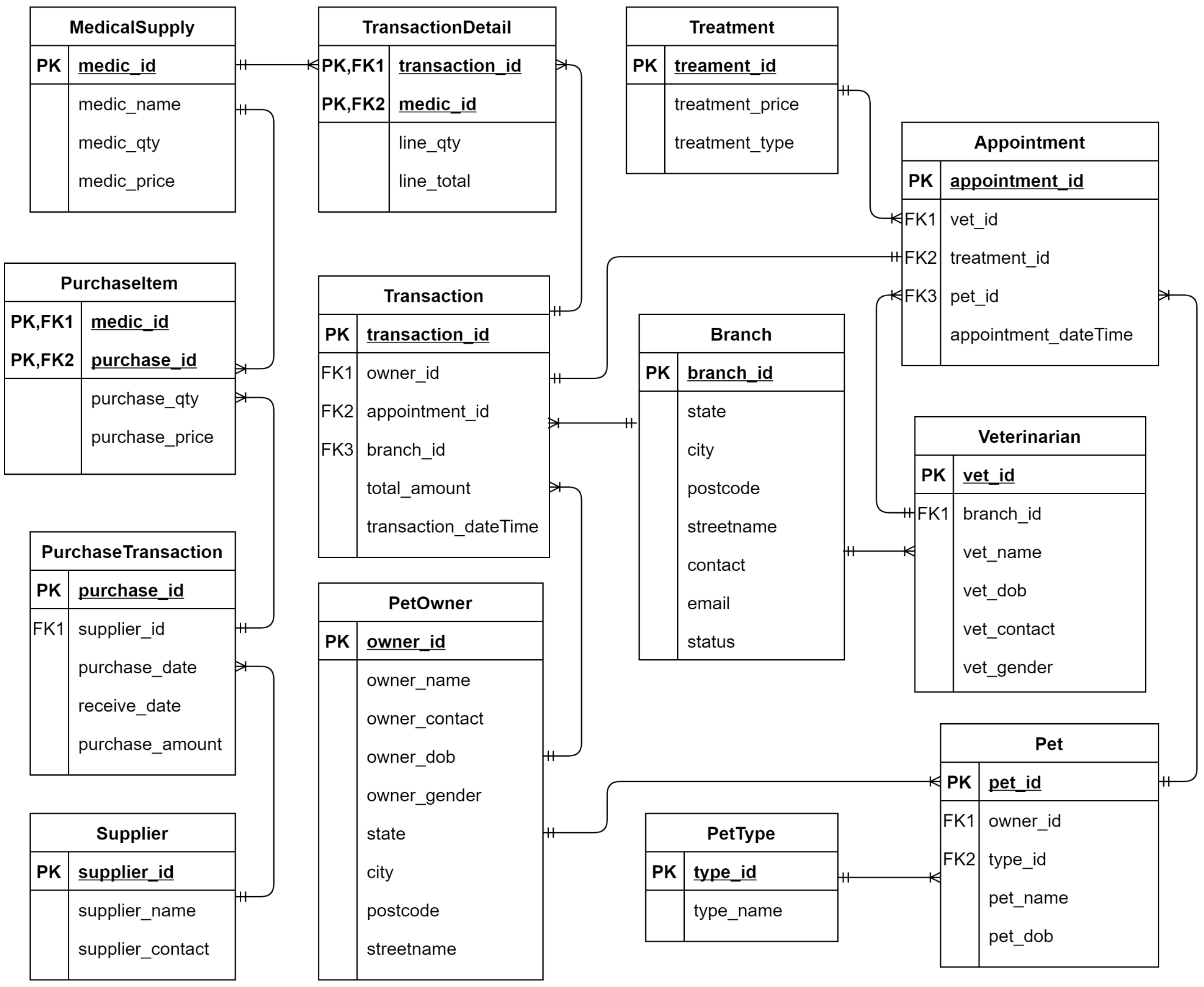
The purpose of this function is to have an easy way to register and record the pet information into the system. By recording the pet detail, it allows the staff to check whether the owner has registered before or not. If not yet registered, the system will require the staff to register the pet owner first before adding the pet.

Medical Stock Management

This function is mainly to manage the stock in the Wellness Veterinary Clinic such as the medicine. It can help us to keep track of the transaction of medicine from the supplier and and the transaction of medicine from the customer. This function is able to calculate the total amount of transactions for the medicine that we use to order from the supplier.

Chapter 2 Entity-Relationship Modeling

2.1 ERD



2.2 Assumptions and Business Rules

Branch

1. All branches of pet clinics have the same operation time which is 10am-6pm.
2. One branch can have many veterinarians, but each veterinarian can only work in one branch.(One-to-many)
3. One branch can have many transactions, but each transaction can only have one branch.(One-to-many)

Treatment

1. One treatment can have many appointments, but one appointment can only have one treatment. (One-to-Many)

Pet

1. One pet can have many appointments, but each appointment can only have one pet.(One-to-Many)
2. Each pet can only be recorded under one pet owner, while one pet owner can have multiple pets.(Many-to-One)

PetOwner

1. Each pet owner can have one or many pets, while one pet can only have one pet owner. (One-to-Many)
2. Each pet owner can have one or more transactions, while each transaction can only have one pet owner recorded. (One-to-Many)

Veterinarian

1. One veterinarian can handle many appointments, but each appointment can only be handled by one veterinarian.(One-to-many)
2. One veterinarian can only in one branch, but one branch can have many veterinarians.(Many-to-One)

PurchaseTransaction

1. One PurchaseTransaction can have many PurchaseItem, but one PurchaseItem can only have one PurchaseTransaction. (One-to-Many)
2. One PurchaseTransaction can only have one Supplier, but one Supplier can have many PurchaseTransaction.

PurchaseItem

1. One PurchaseItem can only have one PurchaseTransaction, but one PurchaseTransaction can have many PurchaseItem. (One-to-One)

   <u>Formula</u>
   transacPurAmt = price * quantity

Supplier

1. One supplier can have many PurchaseTransaction, but one PurchaseTransaction can only have one supplier. (One-to-Many)

Medical Supply

1. One medical supply can be included in many transaction details, but a transaction detail can have only one or none medical supply. (Zero/One-to-Many)
2. Price of the medical supply must be more than its purchase price.

Transaction Detail

1. One transaction detail can have one medical supply, a medical supply can be included in many transaction details.(One-to-Many)
2. One transaction detail can only be included in a transaction , a transaction can be included in many transaction details.(One-to-Many)

3. One transaction must include either one medical supply or treatment.
4. Transaction amount cannot be zero.
5. Transaction details cannot be edited after 7 days from the transaction data.

   Formula
   line_total = medic_price * line_qty

Transaction

1. Each single transaction can have many transaction details, but one transaction detail can only be included in one transaction. (One-to-Many)
2. One transaction can have only one pet owner, but a pet owner can have many transactions. (One-to-Many)
3. Each transaction can only have one appointment. (One-to-One)

   Formula
   total_line_total += line_total
   total_amount = total_line_total + treatment_price (Sum up all the line total and treatment price)

Appointment

1. One appointment only can have one pet, but one pet can have many appointments. (One-to-Many)
2. Appointment can only be made in business hours(10.00am to 5.00pm).
3. One appointment only can have one veterinarian, but one veterinarian can have many appointments. (One-to-Many).
4. One appointment only can have one treatment, but one treatment can be included in many appointments. (One-to-Many).
5. Each appointment can only be included in the transaction.
6. The appointment can only be made if the selected veterinarian is available on the selected period.

Chapter 3 Data Definition

Create table statements with appropriate constraints:

3.1 Branch table

```
CREATE TABLE Branch(
 branch_id       CHAR(5)       NOT NULL,
 state           VARCHAR2(30)  NOT NULL,
 city            VARCHAR2(30)  NOT NULL,
 postcode        NUMBER(5)     NOT NULL,
 streetName      VARCHAR2(50)  NOT NULL,
 contact         VARCHAR2(11)  NOT NULL,
 email           VARCHAR2(30)  NOT NULL,
 status          VARCHAR2(10)  NOT NULL,
 PRIMARY KEY (branch_id),
 CONSTRAINT chk_status CHECK(status IN ('Active','Not Active'))
);
```

3.2 Veterinarian table

```
CREATE TABLE Veterinarian(
 vet_id            CHAR(5)       NOT NULL,
 branch_id         CHAR(5)       NOT NULL,
 vet_name          VARCHAR2(30)  NOT NULL,
 vet_dob           DATE          NOT NULL,
 vet_contact       VARCHAR2(11)  NOT NULL,
 vet_gender        CHAR(1)       NOT NULL,
 PRIMARY KEY (vet_id),
 FOREIGN KEY (branch_id) REFERENCES Branch (branch_id),
 CONSTRAINT chk_vet_gender CHECK(vet_gender IN ('M','F')),
 CONSTRAINT chk_vet_contact
CHECK(REGEXP_LIKE(vet_contact,'^[0-9]+$'))
);
```

3.3 Pet Owner table

```
CREATE TABLE PetOwner(
 owner_id              CHAR(5)            NOT NULL,
 owner_name            VARCHAR2(30)       NOT NULL,
 owner_contact         VARCHAR2(11)       NOT NULL,
 owner_dob             DATE               NOT NULL,
 owner_gender          CHAR(1)            NOT NULL,
 state                 VARCHAR2(30)       NOT NULL,
 city                  VARCHAR2(30)       NOT NULL,
 postcode              VARCHAR2(5)        NOT NULL,
 streetName            VARCHAR2(50)       NOT NULL,
 PRIMARY KEY (owner_id),
 CONSTRAINT chk_owner_gender CHECK(owner_gender IN ('M','F')),
 CONSTRAINT chk_owner_name CHECK(REGEXP_LIKE(owner_name,'^[a-z
A-z]+$')),
 CONSTRAINT chk_owner_contact
CHECK(REGEXP_LIKE(owner_contact,'^[0-9]+$'))
);
```

3.4 Pet Type table

```
CREATE TABLE PetType(
 type_id               CHAR(5)         NOT NULL,
 type_name             VARCHAR2(30)    NOT NULL,
 PRIMARY KEY (type_id),
 CONSTRAINT chk_type_name CHECK(REGEXP_LIKE(type_name,'^[a-z
A-z]+$'))
);
```

3.5 Pet table

```
CREATE TABLE Pet(
 pet_id               CHAR(5)         NOT NULL,
 owner_id             CHAR(5)         NOT NULL,
 pet_name             VARCHAR2(30)    NOT NULL,
 pet_dob              DATE            NOT NULL,
 type_id              CHAR(5)         NOT NULL,
 PRIMARY KEY (pet_id),
 FOREIGN KEY (owner_id) REFERENCES PetOwner (owner_id),
 FOREIGN KEY (type_id) REFERENCES PetType (type_id),
 CONSTRAINT chk_pet_name CHECK(REGEXP_LIKE(pet_name,'^[a-z
A-z]+$'))
);
```

3.6 Treatment table

```
CREATE TABLE Treatment(
 treatment_id      CHAR(5)       NOT NULL,
 treatment_price   NUMBER(7,2)   NOT NULL,
 treatment_type    VARCHAR2(50)  NOT NULL,
 PRIMARY KEY (treatment_id),
 CONSTRAINT chk_treatment_price CHECK(treatment_price > 0)
);
```

3.7 Appointment table

```
CREATE TABLE Appointment(
 appointment_id         CHAR(10)     NOT NULL,
 vet_id                 CHAR(5)      NOT NULL,
 treatment_id           CHAR(5)      NOT NULL,
 pet_id                 CHAR(5)      NOT NULL,
 appointment_dateTime   DATE         NOT NULL,
 PRIMARY KEY (appointment_id),
 FOREIGN KEY (vet_id) REFERENCES Veterinarian (vet_id),
 FOREIGN KEY (treatment_id) REFERENCES Treatment (treatment_id),
 FOREIGN KEY (pet_id) REFERENCES Pet (pet_id)
);
```

3.8 Transaction table

```
CREATE TABLE Transaction(
 transaction_id          CHAR(11)     NOT NULL,
 owner_id                CHAR(5)      NOT NULL,
 appointment_id          CHAR(10)     NOT NULL,
 branch_id               CHAR(5)      NOT NULL,
 total_amount            NUMBER(7,2)  NOT NULL,
 transaction_dateTime    DATE         NOT NULL,
 PRIMARY KEY (transaction_id),
 FOREIGN KEY (owner_id) REFERENCES PetOwner (owner_id),
 FOREIGN KEY (appointment_id) REFERENCES Appointment
(appointment_id),
 FOREIGN KEY (branch_id) REFERENCES Branch (branch_id),
 CONSTRAINT chk_total_amount CHECK(total_amount > 0)
);
```

3.9 Medical Supply table

```
CREATE TABLE MedicalSupply(
 medic_id        CHAR(5)       NOT NULL,
 medic_name      VARCHAR2(30)  NOT NULL,
 medic_qty       NUMBER(5)     NOT NULL,
 medic_price     NUMBER(7,2)   NOT NULL,
 PRIMARY KEY(medic_id),
 CONSTRAINT chk_qty CHECK(medic_qty>=0),
 CONSTRAINT chk_price CHECK(medic_price>0)
);
```

3.10 Transaction Detail table

```
CREATE TABLE TransactionDetail(
 transaction_id    CHAR(11)     NOT NULL,
 medic_id          CHAR(5)      NOT NULL,
 line_qty          NUMBER(3)    NOT NULL,
 line_total        Number(7,2)  NOT NULL,
 PRIMARY KEY (transaction_id, medic_id),
 FOREIGN KEY (transaction_id) REFERENCES Transaction
(transaction_id),
 FOREIGN KEY (medic_id) REFERENCES MedicalSupply (medic_id),
 CONSTRAINT chk_line_total CHECK(line_total > 0),
 CONSTRAINT chk_line_qty CHECK(line_qty > 0)
);
```

3.11 Supplier table

```
CREATE TABLE Supplier(
 supplier_id       CHAR(5)       NOT NULL,
 supplier_name     VARCHAR2(50)  NOT NULL,
 supplier_contact  VARCHAR2(11)  NOT NULL,
 PRIMARY KEY (supplier_id),
 CONSTRAINT chk_sup_name CHECK(REGEXP_LIKE(supplier_name,'^[a-z
A-z]+$')),
 CONSTRAINT chk_sup_contact
CHECK(REGEXP_LIKE(supplier_contact,'^[0-9]+$'))
);
```

3.12 Purchase Transaction table

```
CREATE TABLE PurchaseTransaction(
 purchase_id      CHAR(5)      NOT NULL,
 supplier_id      CHAR(5)      NOT NULL,
 purchase_date    DATE         NOT NULL,
 receive_date     DATE         NOT NULL,
 purchase_amount  NUMBER(7,2)  NOT NULL,
 PRIMARY KEY (purchase_id),
 FOREIGN KEY (supplier_id) REFERENCES Supplier (supplier_id),
 CONSTRAINT chk_pur_amount CHECK(purchase_amount>= 0)
);
```

3.13 Purchase Item table

```
CREATE TABLE PurchaseItem(
 medic_id         CHAR(5)      NOT NULL,
 purchase_id      CHAR(5)      NOT NULL,
 purchase_qty     NUMBER(3)    NOT NULL,
 purchase_price   NUMBER(7,2)  NOT NULL,
 PRIMARY KEY (medic_id, purchase_id),
 FOREIGN KEY (medic_id) REFERENCES MedicalSupply (medic_id),
 FOREIGN KEY (purchase_id) REFERENCES PurchaseTransaction
(purchase_id),
 CONSTRAINT chk_pqty CHECK(purchase_qty>0),
 CONSTRAINT chk_pprice CHECK(purchase_price>0)
);
```

Chapter 4 Queries, Procedures, Triggers and Reports

4.1 (Tan Yi Hong)

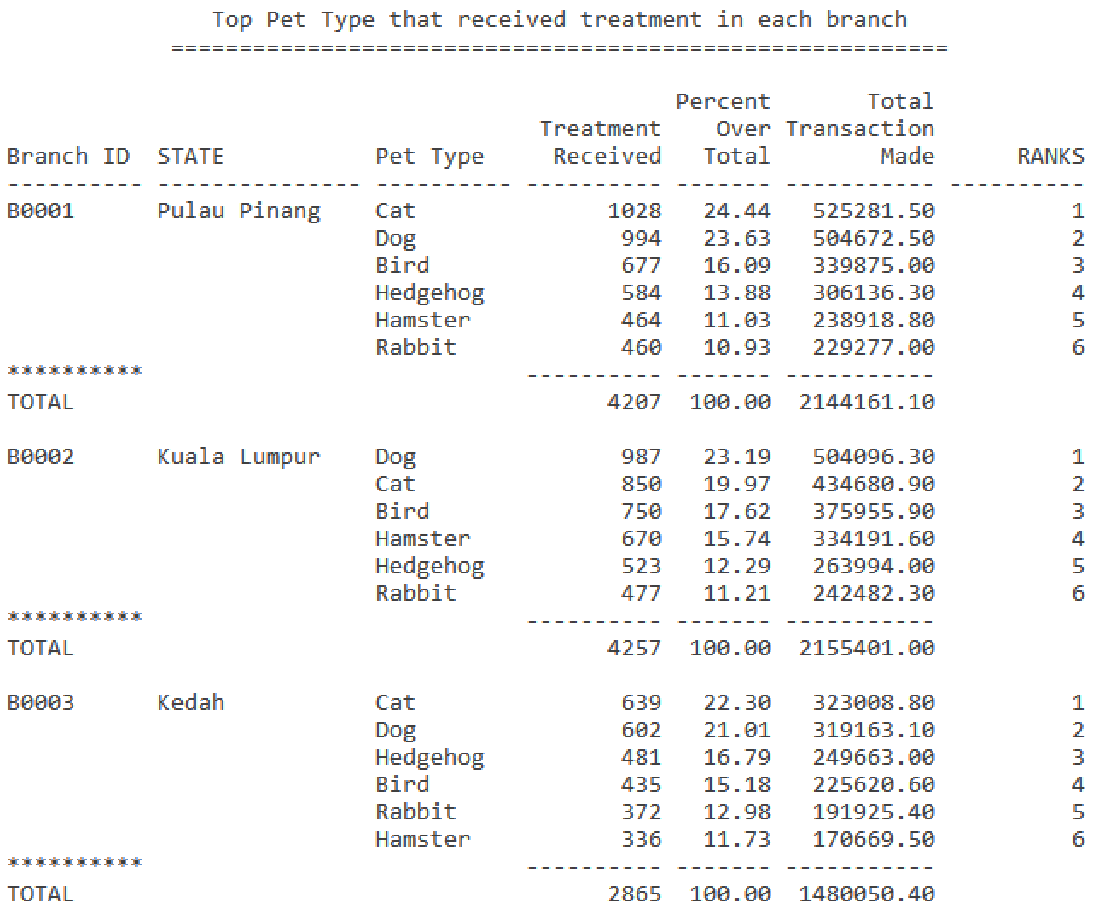
4.1.1 Query 1: Top pet type that received treatment in each branch (Strategic)

Purpose: The purpose of this query is to let the clinic know the top pet type that received treatment in each branch so that the organization can focus more on the service on which type of pets in each branch to provide better service to their customer for all branches.

```
clear break
clear compute
set linesize 80
set pagesize 100
break on state on branch_id skip 1
COMPUTE SUM LABEL TOTAL OF nooftreatment percentage
transactionamount on branch_id
TTITLE ON
TTITLE CENTER  'Top Pet Type that received treatment in each
branch' SKIP 1-
CENTER
==========================================================
SKIP 2
COLUMN branch_id FORMAT a10
COLUMN branch_id HEADING 'Branch ID'
COLUMN state FORMAT a15
COLUMN type_name FORMAT a10
COLUMN type_name HEADING 'Pet Type'
COLUMN nooftreatment HEADING 'Treatment|Received'
COLUMN transactionamount FORMAT 9999999.99
COLUMN transactionamount HEADING 'Total|Transaction|Made'
COLUMN Percentage FORMAT 999.99
COLUMN Percentage HEADING 'Percent|Over|Total'

CREATE OR REPLACE VIEW topPetTreatment AS
SELECT t.branch_id, pt.type_name, COUNT(t.appointment_id) AS
NoOfTreatment, SUM(t.total_amount) AS TransactionAmount
FROM appointment a, veterinarian v, pet p, petType pt,
transaction t
WHERE t.appointment_id=a.appointment_id AND a.vet_id=v.vet_id
AND a.pet_id=p.pet_id
      AND p.type_id=pt.type_id AND
t.appointment_id=a.appointment_id
GROUP BY t.branch_id, pt.type_name
ORDER BY t.branch_id, SUM(t.total_amount) DESC;

SELECT a.branch_id, b.state, a.type_name, a.nooftreatment,
(a.NoOfTreatment/COUNT(t.appointment_id))*100 AS Percentage,
a.transactionAmount, RANK() OVER(PARTITION BY a.branch_id
ORDER BY a.transactionAmount DESC) Ranks
```

```
FROM topPetTreatment a, transaction t, branch b
WHERE a.branch_id=t.branch_id AND a.branch_id=b.branch_id
GROUP BY a.branch_id, b.state, a.type_name, a.nooftreatment,
a.transactionAmount
ORDER BY a.branch_id, a.transactionamount DESC;
```

Sample Output:



```
                 Top Pet Type that received treatment in each branch
               ==========================================================

                                                  Percent       Total
                                        Treatment    Over Transaction
Branch ID  STATE           Pet Type      Received   Total        Made       RANKS
---------- --------------- ---------- ---------- ------- ----------- ----------
B0001      Pulau Pinang    Cat              1028   24.44   525281.50          1
                           Dog               994   23.63   504672.50          2
                           Bird              677   16.09   339875.00          3
                           Hedgehog          584   13.88   306136.30          4
                           Hamster           464   11.03   238918.80          5
                           Rabbit            460   10.93   229277.00          6
**********                                ---------- ------- -----------
TOTAL                                           4207  100.00  2144161.10

B0002      Kuala Lumpur    Dog               987   23.19   504096.30          1
                           Cat               850   19.97   434680.90          2
                           Bird              750   17.62   375955.90          3
                           Hamster           670   15.74   334191.60          4
                           Hedgehog          523   12.29   263994.00          5
                           Rabbit            477   11.21   242482.30          6
**********                                ---------- ------- -----------
TOTAL                                           4257  100.00  2155401.00

B0003      Kedah           Cat               639   22.30   323008.80          1
                           Dog               602   21.01   319163.10          2
                           Hedgehog          481   16.79   249663.00          3
                           Bird              435   15.18   225620.60          4
                           Rabbit            372   12.98   191925.40          5
                           Hamster           336   11.73   170669.50          6
**********                                ---------- ------- -----------
TOTAL                                           2865  100.00  1480050.40
```

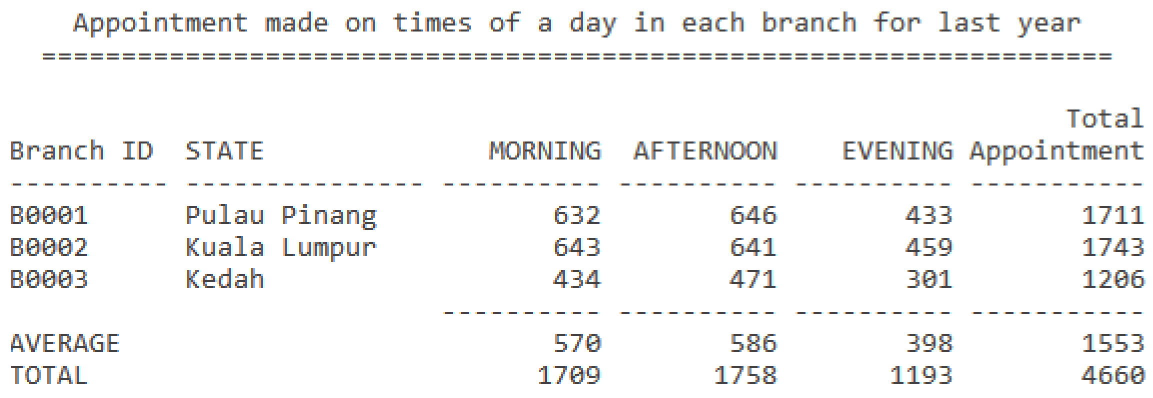
4.1.2 Query 2: Appointment made on times of a day in each branch for last year (Tactical)

Purpose: The purpose of this query is to list out all appointments made during the times of a day in each branch and to know the peak business time to let the organization adjust the shifts between the veterinarian to suitable shifts whether increasing or decreasing the time shifts between them.

```
clear break
clear compute
set linesize 71
set pagesize 100
BREAK ON REPORT
COMPUTE SUM LABEL TOTAL AVG LABEL AVERAGE OF MORNING
AFTERNOON EVENING totalappointment ON REPORT
COLUMN morning FORMAT 999999999
COLUMN afternoon FORMAT 999999999
COLUMN evening FORMAT 999999999
TTITLE ON
TTITLE CENTER 'Appointment made on times of a day in each
branch for last year' SKIP 1-
CENTER
=============================================================
====== SKIP 2
COLUMN branch_id FORMAT a10
COLUMN branch_id HEADING 'Branch ID'
COLUMN state FORMAT a15
COLUMN totalappointment FORMAT 999999999
COLUMN totalappointment HEADING 'Total|Appointment'

CREATE OR REPLACE VIEW morningApp AS
SELECT t.branch_id, COUNT(t.transaction_id) AS MORNING
FROM appointment a, transaction t
WHERE t.appointment_id = a.appointment_id AND
      EXTRACT(HOUR FROM CAST(a.appointment_datetime AS
TIMESTAMP)) BETWEEN 10 AND 12
              AND EXTRACT(YEAR FROM a.appointment_datetime) =
EXTRACT(YEAR FROM SYSDATE)-1
GROUP BY t.branch_id
ORDER BY t.branch_id;

CREATE OR REPLACE VIEW afternoonApp AS
SELECT t.branch_id, COUNT(t.transaction_id) AS AFTERNOON
FROM appointment a, transaction t
WHERE t.appointment_id = a.appointment_id AND
      EXTRACT(HOUR FROM CAST(a.appointment_datetime AS
TIMESTAMP)) BETWEEN 13 AND 15
              AND EXTRACT(YEAR FROM a.appointment_datetime) =
EXTRACT(YEAR FROM SYSDATE)-1
GROUP BY t.branch_id
ORDER BY t.branch_id;

CREATE OR REPLACE VIEW eveningApp AS
SELECT t.branch_id, COUNT(t.transaction_id) AS EVENING
FROM appointment a, transaction t
WHERE t.appointment_id = a.appointment_id AND
      EXTRACT(HOUR FROM CAST(a.appointment_datetime AS
TIMESTAMP)) BETWEEN 16 AND 18
              AND EXTRACT(YEAR FROM a.appointment_datetime) =
EXTRACT(YEAR FROM SYSDATE)-1
GROUP BY t.branch_id
ORDER BY t.branch_id;

SELECT a.branch_id, d.state, a.morning, b.afternoon,
c.evening, (a.morning + b.afternoon + c.evening) AS
TotalAppointment
FROM morningApp a, afternoonApp b, eveningApp c, branch d
```

```
WHERE a.branch_id=b.branch_id AND a.branch_id=c.branch_id AND
a.branch_id=d.branch_id
GROUP BY a.branch_id, d.state, a.morning, b.afternoon,
c.evening
ORDER BY a.branch_id;
```

Sample Output:



```
    Appointment made on times of a day in each branch for last year
  ==================================================================

                                                                Total
Branch ID  STATE                MORNING  AFTERNOON    EVENING Appointment
---------- --------------- ---------- ---------- ---------- -----------
B0001      Pulau Pinang           632        646        433        1711
B0002      Kuala Lumpur           643        641        459        1743
B0003      Kedah                  434        471        301        1206
                           ---------- ---------- ---------- -----------
AVERAGE                           570        586        398        1553
TOTAL                            1709       1758       1193        4660
```

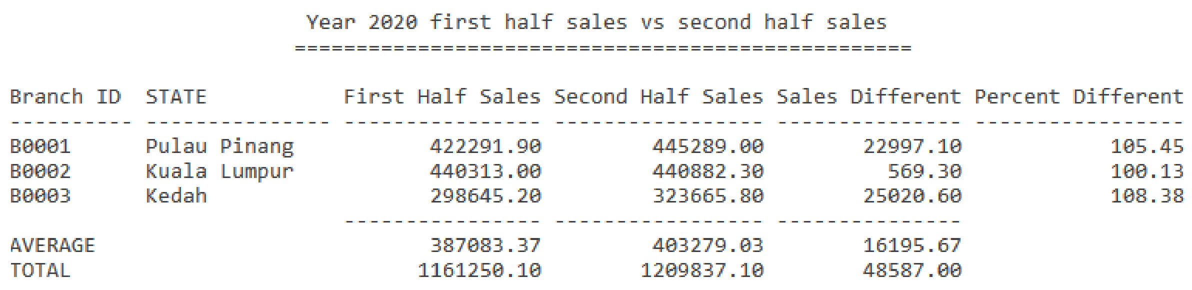
4.1.3 Query 3: Year 2020 first half sales vs second half sales in each branch (Operational)

Purpose: The purpose of this query is to calculate and compare the sales of first half and second half in the year 2020 for each branch. It can show the operational sales of the clinic and compare the sales difference or percent difference to better understand the trends of their sales growth.

```
clear break
clear compute
BREAK ON REPORT
COMPUTE SUM LABEL TOTAL AVG LABEL AVERAGE OF
SALES2020_1STHALF SALES2020_2NDHALF SALESDIFF ON REPORT
set linesize 95
set pagesize 100
TTITLE ON
TTITLE CENTER 'Year 2020 first half sales vs second half
sales' SKIP 1-
CENTER ==================================================
SKIP 2
COLUMN branch_id FORMAT a10
COLUMN branch_id HEADING 'Branch ID'
COLUMN state FORMAT a15
COLUMN SALES2020_1STHALF FORMAT 9999999.99
COLUMN SALES2020_1STHALF HEADING 'First Half Sales'
COLUMN SALES2020_2NDHALF FORMAT 9999999.99
COLUMN SALES2020_2NDHALF HEADING 'Second Half Sales'
COLUMN SALESDIFF FORMAT 9999999.99
COLUMN SALESDIFF HEADING 'Sales Different'
COLUMN SALESDIFF_PERCENTAGE FORMAT 999.99
COLUMN SALESDIFF_PERCENTAGE HEADING 'Percent Different'
```

```
CREATE OR REPLACE VIEW Sales2020_1stHalf AS
SELECT branch_id, SUM(total_amount) AS Sales2020_1stHalf
FROM transaction
WHERE EXTRACT(YEAR FROM transaction_dateTime) = 2020 AND
EXTRACT(MONTH FROM transaction_dateTime) <= 6
GROUP BY branch_id
ORDER BY branch_id, SUM(total_amount) DESC;

CREATE OR REPLACE VIEW Sales2020_2ndHalf AS
SELECT branch_id, SUM(total_amount) AS Sales2020_2ndHalf
FROM transaction
WHERE EXTRACT(YEAR FROM transaction_dateTime) = 2020 AND
EXTRACT(MONTH FROM transaction_dateTime) > 6
GROUP BY branch_id
ORDER BY branch_id, SUM(total_amount) DESC;

SELECT a.branch_id, c.state, a.Sales2020_1stHalf,
b.Sales2020_2ndHalf, Sales2020_2ndHalf-Sales2020_1stHalf AS
SalesDiff, (Sales2020_2ndHalf/Sales2020_1stHalf)*100 AS
SalesDiff_Percentage
FROM branch c, Sales2020_1stHalf a, Sales2020_2ndHalf b
WHERE a.branch_id=c.branch_id AND b.branch_id=c.branch_id
GROUP BY a.branch_id, c.state, a.Sales2020_1stHalf,
b.Sales2020_2ndHalf
ORDER BY branch_id;
```

Sample Output:



```
                              Year 2020 first half sales vs second half sales
                             ==================================================

Branch ID  STATE           First Half Sales Second Half Sales Sales Different Percent Different
---------- --------------- ---------------- ----------------- --------------- -----------------
B0001      Pulau Pinang           422291.90         445289.00        22997.10            105.45
B0002      Kuala Lumpur           440313.00         440882.30          569.30            100.13
B0003      Kedah                  298645.20         323665.80        25020.60            108.38
                           ---------------- ----------------- ---------------
AVERAGE                           387083.37         403279.03        16195.67
TOTAL                            1161250.10        1209837.10        48587.00
```

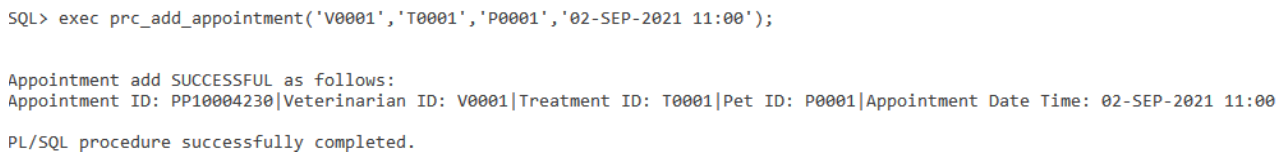
4.1.4 Procedure 1: Add Appointment record

Purpose: The purpose of this stored procedure is to add a new appointment record into the database when customers want to make new appointments.

```
CREATE OR REPLACE PROCEDURE PRC_ADD_APPOINTMENT(IN_vetID in
CHAR, IN_treatmentID in CHAR, IN_petID in CHAR, IN_dateTime
in DATE) AS
   v_insertID   CHAR(10);
   v_branchID   CHAR(5);
   counter_t    NUMBER;
   counter_p    NUMBER;
   e_invalid_treatment EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_invalid_treatment, -20050);
   e_invalid_pet EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_invalid_pet, -20051);

BEGIN
   counter_t := 0;
   counter_p := 0;

   SELECT branch_id INTO v_branchID
   FROM veterinarian
   WHERE vet_id = IN_vetID;

   SELECT COUNT(*) INTO counter_t
   FROM treatment
   WHERE treatment_id = IN_treatmentID;

   IF counter_t = 0 THEN
      RAISE_APPLICATION_ERROR(-20050, 'Invalid Treatment
ID.');
   END IF;

   SELECT COUNT(*) INTO counter_p
   FROM pet
   WHERE pet_id = IN_petID;

   IF counter_p = 0 THEN
      RAISE_APPLICATION_ERROR(-20051, 'Invalid Pet ID.');
   END IF;

   IF v_branchID = 'B0001' THEN
      v_insertID := TO_CHAR('PP'||app_seq_PG.NEXTVAL);

   ELSIF v_branchID = 'B0002' THEN
      v_insertID := TO_CHAR('KL'||app_seq_KL.NEXTVAL);

   ELSIF v_branchID = 'B0003' THEN
      v_insertID := TO_CHAR('KD'||app_seq_KD.NEXTVAL);

   END IF;

   insert into appointment
values(v_insertID,IN_vetID,IN_treatmentID,IN_petID,IN_dateTim
e);

   DBMS_OUTPUT.PUT_LINE (CHR(10));
   DBMS_OUTPUT.PUT_LINE ('Appointment add SUCCESSFUL as
follows: ');
   DBMS_OUTPUT.PUT_LINE ('Appointment ID:
'||v_insertID||'|Veterinarian ID: '||IN_vetID||'|Treatment
ID: '||IN_treatmentID||'|Pet ID: '||IN_petID||'|Appointment
Date Time: '||IN_dateTime);
```

```
 EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE ('No Veterinarian found');
   WHEN e_invalid_treatment THEN
      DBMS_OUTPUT.PUT_LINE('No such Treatment ID');
                                 DBMS_OUTPUT.PUT_LINE(SQLERRM);
   WHEN e_invalid_pet THEN
      DBMS_OUTPUT.PUT_LINE('No such Pet ID');
                                 DBMS_OUTPUT.PUT_LINE(SQLERRM);

END;
/
```

Sample Output:



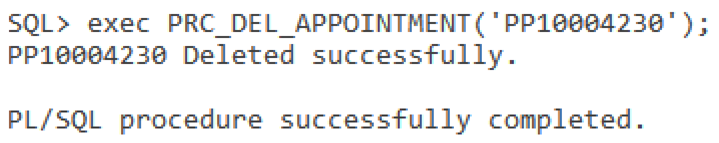
```
SQL> exec prc_add_appointment('V0001','T0001','P0001','02-SEP-2021 11:00');


Appointment add SUCCESSFUL as follows:
Appointment ID: PP10004230|Veterinarian ID: V0001|Treatment ID: T0001|Pet ID: P0001|Appointment Date Time: 02-SEP-2021 11:00

PL/SQL procedure successfully completed.
```

4.1.5 Procedure 2: Delete Appointment record

Purpose: The purpose of this stored procedure is to delete an existing record from the appointment table when a customer has cancelled the appointment made. It will only require the appointment ID to delete the record.

```
CREATE OR REPLACE PROCEDURE
PRC_DEL_APPOINTMENT(IN_appointmentID in CHAR) AS

BEGIN
   DELETE FROM appointment
   WHERE appointment_id = IN_appointmentID;

   DBMS_OUTPUT.PUT_LINE (IN_appointmentID||' Deleted
successfully.');

   EXCEPTION
      WHEN NO_DATA_FOUND THEN
         DBMS_OUTPUT.PUT_LINE ('No Appointment found');

END;
/
```

Sample Output:



```
SQL> exec PRC_DEL_APPOINTMENT('PP10004230');
PP10004230 Deleted successfully.

PL/SQL procedure successfully completed.
```

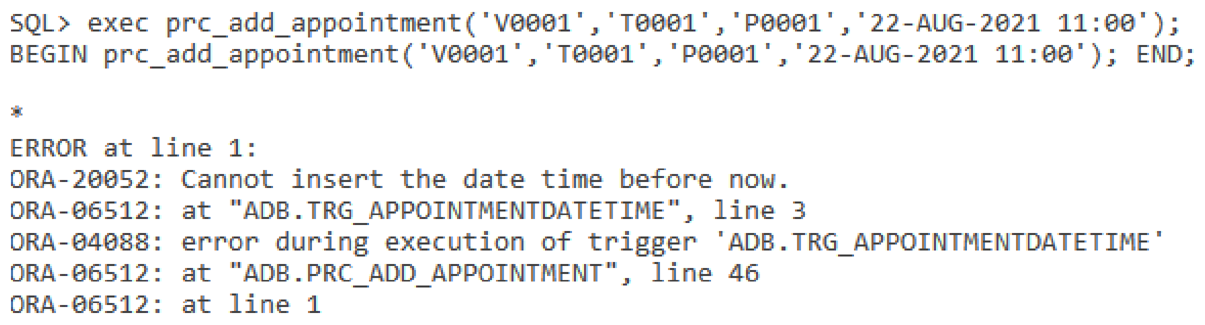
4.1.6 Trigger 1: Validate the proper date time on insertion of appointment

Purpose: The purpose of this trigger is to validate the newly inserted appointment to match in a proper date and time which can make an appointment.

```
CREATE OR REPLACE TRIGGER trg_appointmentDateTime
  BEFORE INSERT OR UPDATE ON Appointment
  FOR EACH ROW
BEGIN
  IF :new.appointment_datetime<SYSDATE THEN
    RAISE_APPLICATION_ERROR(-20052, 'Cannot insert the date
time before now.' );
  ELSIF EXTRACT(HOUR FROM CAST(:new.appointment_datetime AS
TIMESTAMP)) < 10 THEN
    RAISE_APPLICATION_ERROR(-20053, 'Date time must be after
business hour.' );
  ELSIF EXTRACT(HOUR FROM CAST(:new.appointment_datetime AS
TIMESTAMP)) > 17 THEN
    RAISE_APPLICATION_ERROR(-20054, 'Date time must be before
business hour.' );
  END IF;
END;
/
```

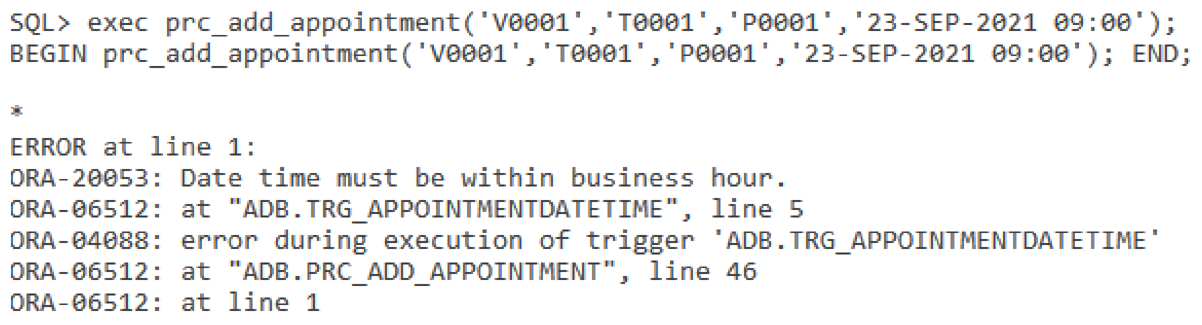
Sample Output:

When the appointment insert is before now:



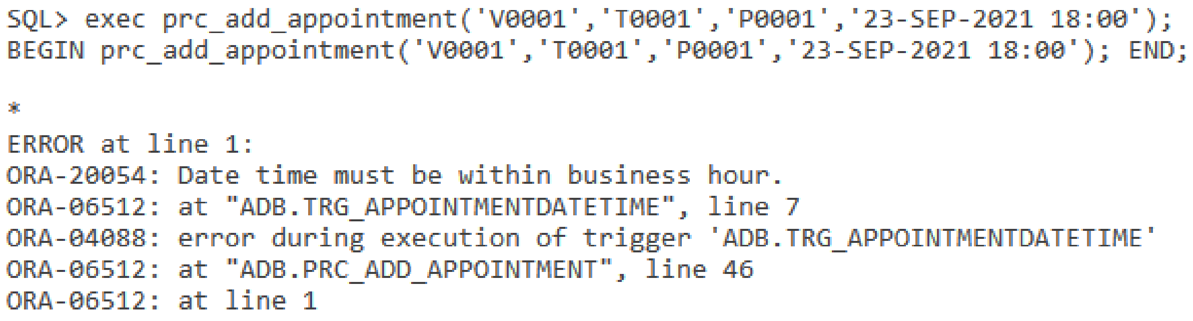
```
SQL> exec prc_add_appointment('V0001','T0001','P0001','22-AUG-2021 11:00');
BEGIN prc_add_appointment('V0001','T0001','P0001','22-AUG-2021 11:00'); END;

*
ERROR at line 1:
ORA-20052: Cannot insert the date time before now.
ORA-06512: at "ADB.TRG_APPOINTMENTDATETIME", line 3
ORA-04088: error during execution of trigger 'ADB.TRG_APPOINTMENTDATETIME'
ORA-06512: at "ADB.PRC_ADD_APPOINTMENT", line 46
ORA-06512: at line 1
```

When the appointment insert is before the business hour:



```
SQL> exec prc_add_appointment('V0001','T0001','P0001','23-SEP-2021 09:00');
BEGIN prc_add_appointment('V0001','T0001','P0001','23-SEP-2021 09:00'); END;

*
ERROR at line 1:
ORA-20053: Date time must be within business hour.
ORA-06512: at "ADB.TRG_APPOINTMENTDATETIME", line 5
ORA-04088: error during execution of trigger 'ADB.TRG_APPOINTMENTDATETIME'
ORA-06512: at "ADB.PRC_ADD_APPOINTMENT", line 46
ORA-06512: at line 1
```

When the appointment insert is after the business hour:



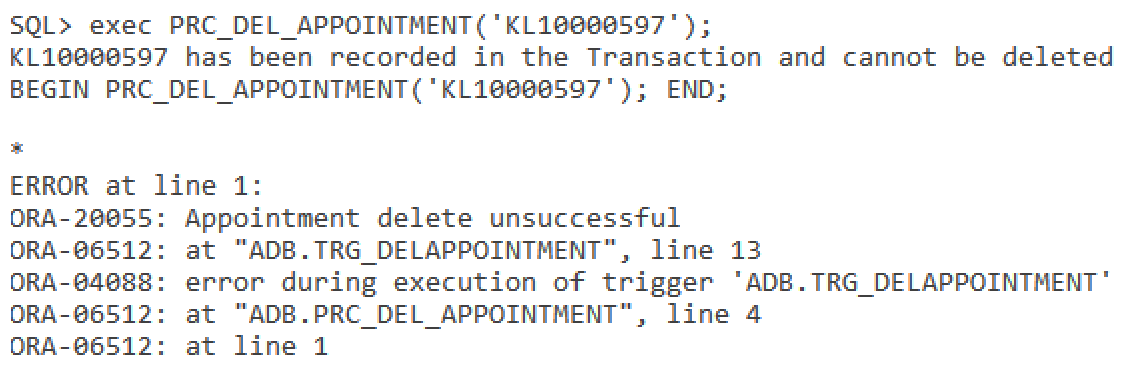
```
SQL> exec prc_add_appointment('V0001','T0001','P0001','23-SEP-2021 18:00');
BEGIN prc_add_appointment('V0001','T0001','P0001','23-SEP-2021 18:00'); END;

*
ERROR at line 1:
ORA-20054: Date time must be within business hour.
ORA-06512: at "ADB.TRG_APPOINTMENTDATETIME", line 7
ORA-04088: error during execution of trigger 'ADB.TRG_APPOINTMENTDATETIME'
ORA-06512: at "ADB.PRC_ADD_APPOINTMENT", line 46
ORA-06512: at line 1
```

4.1.7 Trigger 2: Monitor the deletion of appointment record

Purpose: The purpose of this trigger is to check whether the appointment that user wants to delete is recorded in the transaction or not, and it will be unable to delete when the record is in the transaction already.

```
CREATE OR REPLACE TRIGGER trg_delAppointment
   BEFORE DELETE ON Appointment
   FOR EACH ROW

DECLARE
   counter  NUMBER;

BEGIN
   counter := 0;

   SELECT COUNT(*) INTO counter
   FROM transaction
   WHERE appointment_id = :old.appointment_id;

   IF counter = 1 THEN
      DBMS_OUTPUT.PUT_LINE(:old.appointment_id||' has been
recorded in the Transaction and cannot be deleted');
           RAISE_APPLICATION_ERROR(-20055,'Appointment delete
unsuccessful');
   END IF;

END;
/
```

Sample Output:



```
SQL> exec PRC_DEL_APPOINTMENT('KL10000597');
KL10000597 has been recorded in the Transaction and cannot be deleted
BEGIN PRC_DEL_APPOINTMENT('KL10000597'); END;

*
ERROR at line 1:
ORA-20055: Appointment delete unsuccessful
ORA-06512: at "ADB.TRG_DELAPPOINTMENT", line 13
ORA-04088: error during execution of trigger 'ADB.TRG_DELAPPOINTMENT'
ORA-06512: at "ADB.PRC_DEL_APPOINTMENT", line 4
ORA-06512: at line 1
```

4.1.8 Report 1: Summary report of Specific veterinarian with all of his/her transactions done in a year

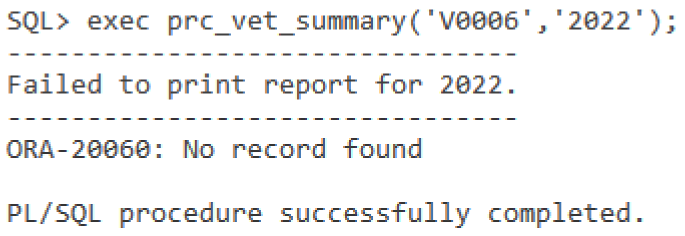
Purpose: The purpose of this report is to summarize a specific veterinarian with all of his/her transactions done in a selected year with the form of a detailed report. It can let organizations know the contribution of the transaction amount they made or to trace the total amount of transactions done by that veterinarian.

```
CREATE OR REPLACE PROCEDURE prc_vet_summary(IN_vetID in CHAR,
IN_year in NUMBER) AS

   v_vetName     VARCHAR2(50);
   v_branchID    CHAR(5);
   v_state       VARCHAR2(50);
   v_totalAmount NUMBER;
   counter       NUMBER;
   record_count  NUMBER;
   e_norecord    EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_norecord,-20060);

   CURSOR vet_trans IS
      SELECT  t.transaction_id, t.appointment_id,
tr.treatment_type, t.transaction_dateTime, t.total_amount
              FROM transaction t, appointment a, treatment tr
                 WHERE t.appointment_id=a.appointment_id AND
a.treatment_id=tr.treatment_id AND a.vet_id=IN_vetID AND
EXTRACT(YEAR FROM t.transaction_dateTime) = IN_year
                          ORDER BY transaction_dateTime DESC;

BEGIN
   v_totalAmount := 0;
   counter := 0;
   record_count := 0;

   SELECT COUNT(*) INTO record_count
   FROM transaction t, appointment a
   WHERE t.appointment_id=a.appointment_id AND EXTRACT(YEAR
FROM t.transaction_dateTime) = IN_year AND a.vet_id=IN_vetID;
```

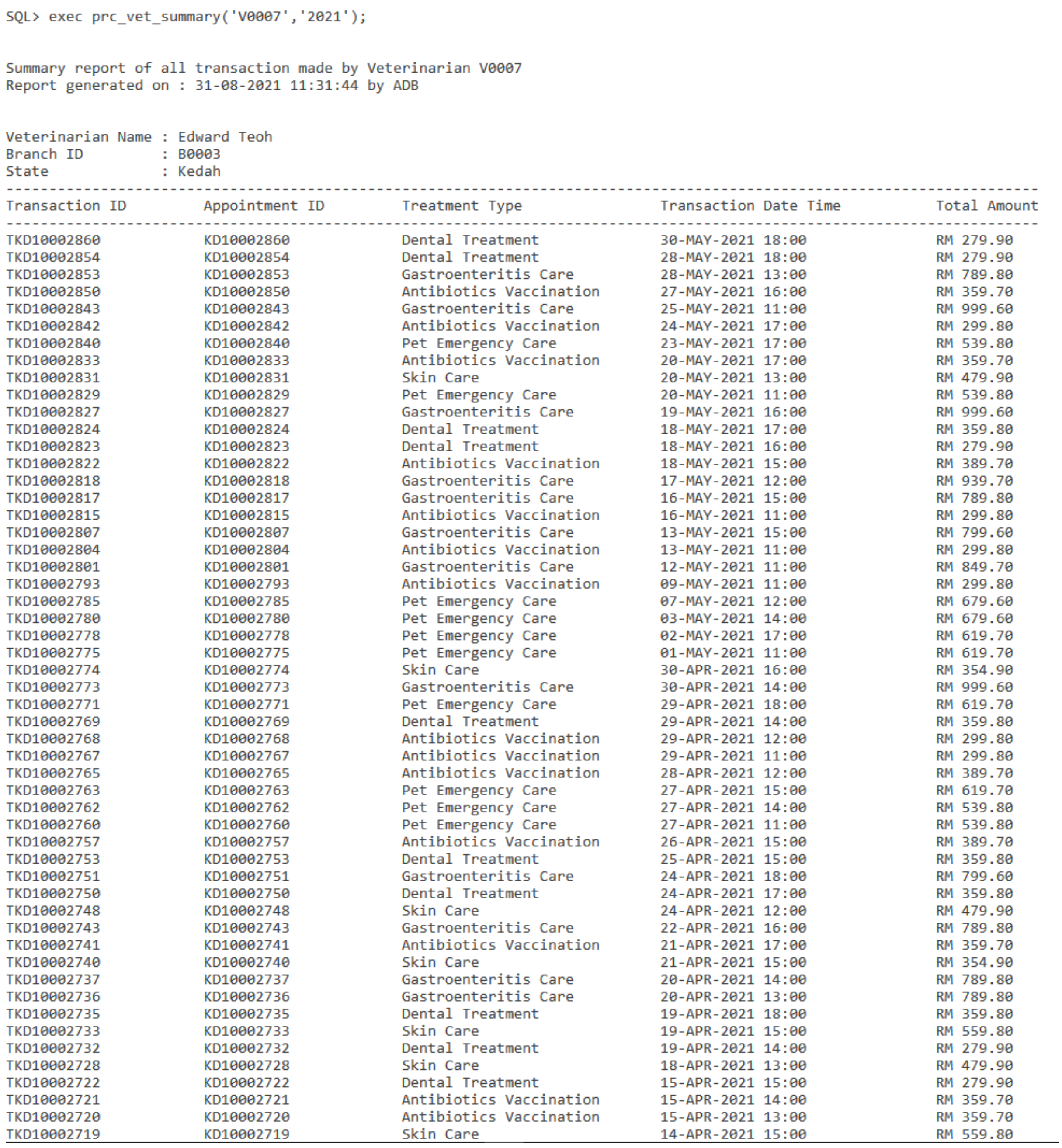
```
   IF record_count = 0 THEN
     RAISE_APPLICATION_ERROR(-20060,'No record found');
   END IF;

   SELECT v.vet_name, v.branch_id, b.state INTO v_vetName,
v_branchID, v_state
   FROM veterinarian v, branch b
   WHERE v.branch_id=b.branch_id AND vet_id=IN_vetID;

   DBMS_OUTPUT.PUT_LINE (chr(10));
   DBMS_OUTPUT.PUT_LINE ('Summary report of all transaction
made by Veterinarian '||IN_vetID);
   DBMS_OUTPUT.PUT_LINE('Report generated on : ' ||
TO_CHAR(CURRENT_DATE, 'DD-MM-YYYY HH:MI:SS') || ' by ' ||
USER);
   DBMS_OUTPUT.PUT_LINE (chr(10));
   DBMS_OUTPUT.PUT_LINE ('Veterinarian Name : '||v_vetName);
   DBMS_OUTPUT.PUT_LINE ('Branch ID         : '||v_branchID);
   DBMS_OUTPUT.PUT_LINE ('State             : '||v_state);

   DBMS_OUTPUT.PUT_LINE(LPAD('-', 120, '-'));
   DBMS_OUTPUT.PUT_LINE(RPAD('Transaction ID', 23, ' ') ||
RPAD('Appointment ID', 23, ' ') || RPAD('Treatment Type', 30,
' ')|| RPAD('Transaction Date Time', 32, ' ') || RPAD('Total
Amount', 20, ' '));
   DBMS_OUTPUT.PUT_LINE(LPAD('-', 120, '-'));

   FOR trans IN vet_trans LOOP
      DBMS_OUTPUT.PUT_LINE(RPAD(trans.transaction_id,23,'
')||RPAD(trans.appointment_id,23,' ')||
RPAD(trans.treatment_type, 30, '
')||RPAD(trans.transaction_dateTime, 32, ' ') ||'RM '||
RPAD(TRIM(TO_CHAR(trans.total_amount,'999G999D99')), 17, '
'));

         v_totalAmount := v_totalAmount + trans.total_amount;
                                     counter := counter + 1;

   END LOOP;

   DBMS_OUTPUT.PUT_LINE(LPAD('=', 120, '='));
   DBMS_OUTPUT.PUT_LINE(RPAD(('Total Treatment Done :
'||counter),75,' ')||'Total Amount of Transaction : RM
'||TRIM(TO_CHAR(v_totalAmount,'999G999D99')));
   DBMS_OUTPUT.PUT_LINE(LPAD('=', 120, '='));

 EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE ('No Veterinarian found');
   WHEN e_norecord THEN
```

```
DBMS_OUTPUT.PUT_LINE('--------------------------------');
      DBMS_OUTPUT.PUT_LINE('Failed to print report for ' ||
IN_year || '.');

DBMS_OUTPUT.PUT_LINE('--------------------------------');
      DBMS_OUTPUT.PUT_LINE(SQLERRM);

END;
/

exec prc_vet_summary('V0007','2021');
```

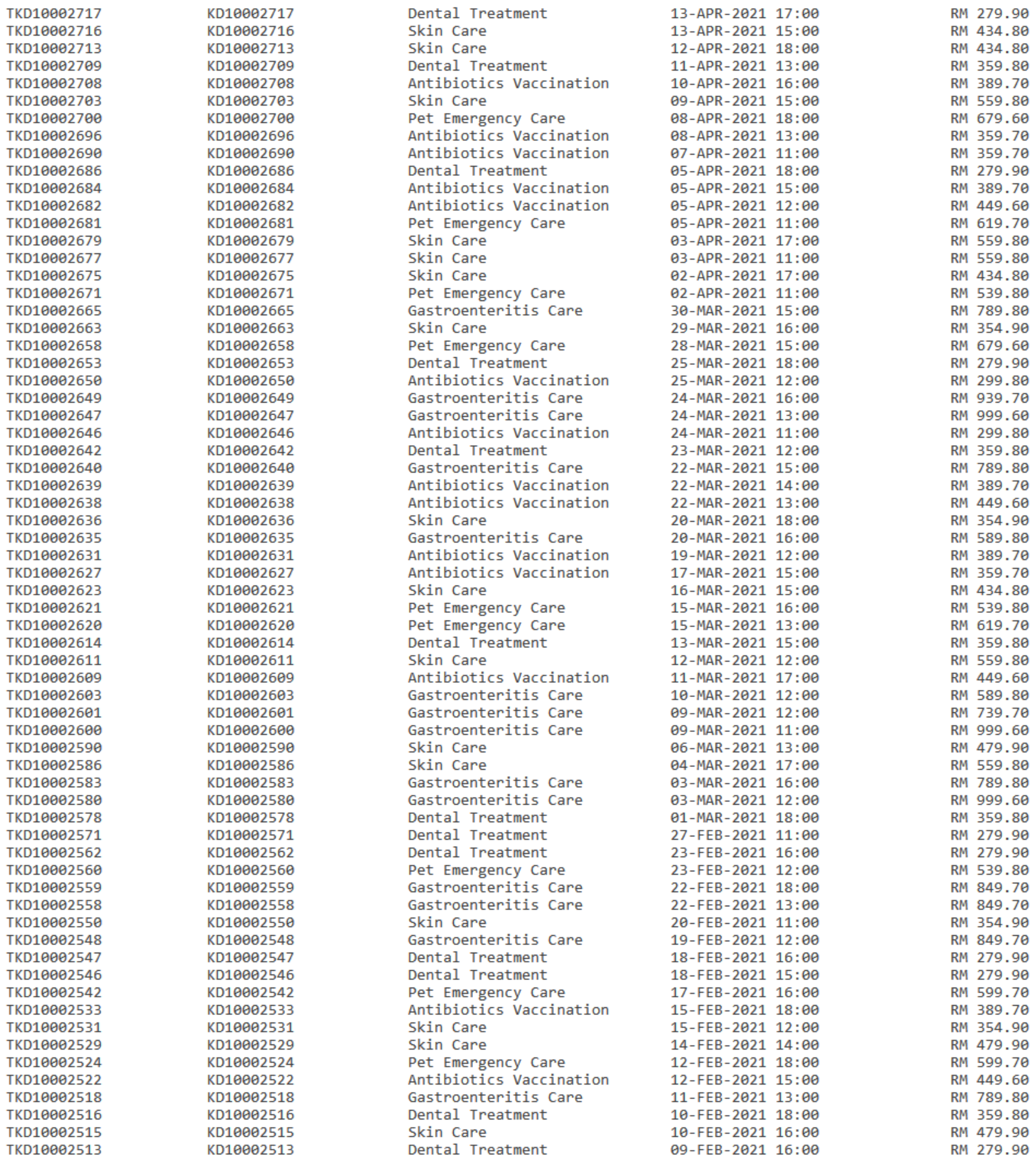
Sample Output:



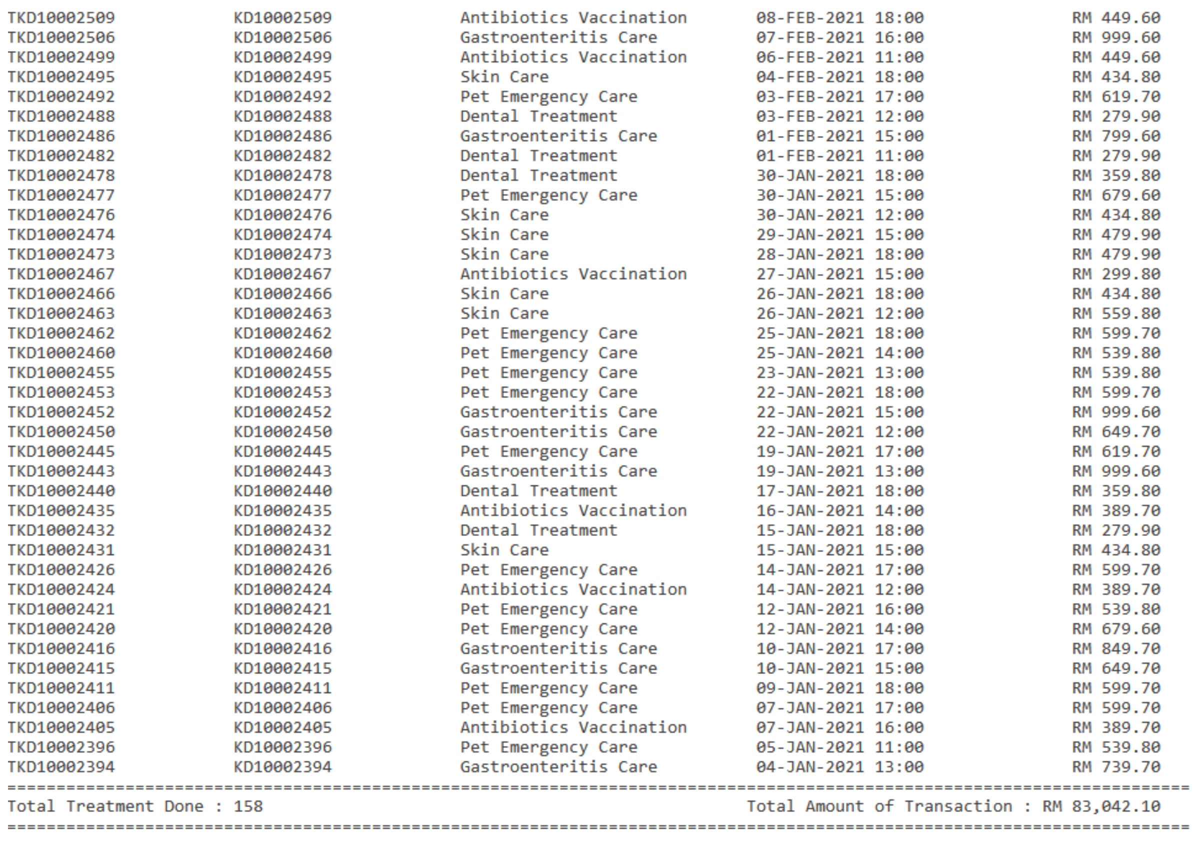
```
SQL> exec prc_vet_summary('V0006','2022');
--------------------------------
Failed to print report for 2022.
--------------------------------
ORA-20060: No record found

PL/SQL procedure successfully completed.
```



```
SQL> exec prc_vet_summary('V0007','2021');


Summary report of all transaction made by Veterinarian V0007
Report generated on : 31-08-2021 11:31:44 by ADB


Veterinarian Name : Edward Teoh
Branch ID         : B0003
State             : Kedah
------------------------------------------------------------------------------------------------------------------------
Transaction ID         Appointment ID         Treatment Type                Transaction Date Time           Total Amount
------------------------------------------------------------------------------------------------------------------------
TKD10002860            KD10002860             Dental Treatment              30-MAY-2021 18:00               RM 279.90
TKD10002854            KD10002854             Dental Treatment              28-MAY-2021 18:00               RM 279.90
TKD10002853            KD10002853             Gastroenteritis Care          28-MAY-2021 13:00               RM 789.80
TKD10002850            KD10002850             Antibiotics Vaccination       27-MAY-2021 16:00               RM 359.70
TKD10002843            KD10002843             Gastroenteritis Care          25-MAY-2021 11:00               RM 999.60
TKD10002842            KD10002842             Antibiotics Vaccination       24-MAY-2021 17:00               RM 299.80
TKD10002840            KD10002840             Pet Emergency Care            23-MAY-2021 17:00               RM 539.80
TKD10002833            KD10002833             Antibiotics Vaccination       20-MAY-2021 17:00               RM 359.70
TKD10002831            KD10002831             Skin Care                     20-MAY-2021 13:00               RM 479.90
TKD10002829            KD10002829             Pet Emergency Care            20-MAY-2021 11:00               RM 539.80
TKD10002827            KD10002827             Gastroenteritis Care          19-MAY-2021 16:00               RM 999.60
TKD10002824            KD10002824             Dental Treatment              18-MAY-2021 17:00               RM 359.80
TKD10002823            KD10002823             Dental Treatment              18-MAY-2021 16:00               RM 279.90
TKD10002822            KD10002822             Antibiotics Vaccination       18-MAY-2021 15:00               RM 389.70
TKD10002818            KD10002818             Gastroenteritis Care          17-MAY-2021 12:00               RM 939.70
TKD10002817            KD10002817             Gastroenteritis Care          16-MAY-2021 15:00               RM 789.80
TKD10002815            KD10002815             Antibiotics Vaccination       16-MAY-2021 11:00               RM 299.80
TKD10002807            KD10002807             Gastroenteritis Care          13-MAY-2021 15:00               RM 799.60
TKD10002804            KD10002804             Antibiotics Vaccination       13-MAY-2021 11:00               RM 299.80
TKD10002801            KD10002801             Gastroenteritis Care          12-MAY-2021 11:00               RM 849.70
TKD10002793            KD10002793             Antibiotics Vaccination       09-MAY-2021 11:00               RM 299.80
TKD10002785            KD10002785             Pet Emergency Care            07-MAY-2021 12:00               RM 679.60
TKD10002780            KD10002780             Pet Emergency Care            03-MAY-2021 14:00               RM 679.60
TKD10002778            KD10002778             Pet Emergency Care            02-MAY-2021 17:00               RM 619.70
TKD10002775            KD10002775             Pet Emergency Care            01-MAY-2021 11:00               RM 619.70
TKD10002774            KD10002774             Skin Care                     30-APR-2021 16:00               RM 354.90
TKD10002773            KD10002773             Gastroenteritis Care          30-APR-2021 14:00               RM 999.60
TKD10002771            KD10002771             Pet Emergency Care            29-APR-2021 18:00               RM 619.70
TKD10002769            KD10002769             Dental Treatment              29-APR-2021 14:00               RM 359.80
TKD10002768            KD10002768             Antibiotics Vaccination       29-APR-2021 12:00               RM 299.80
TKD10002767            KD10002767             Antibiotics Vaccination       29-APR-2021 11:00               RM 299.80
TKD10002765            KD10002765             Antibiotics Vaccination       28-APR-2021 12:00               RM 389.70
TKD10002763            KD10002763             Pet Emergency Care            27-APR-2021 15:00               RM 619.70
TKD10002762            KD10002762             Pet Emergency Care            27-APR-2021 14:00               RM 539.80
TKD10002760            KD10002760             Pet Emergency Care            27-APR-2021 11:00               RM 539.80
TKD10002757            KD10002757             Antibiotics Vaccination       26-APR-2021 15:00               RM 389.70
TKD10002753            KD10002753             Dental Treatment              25-APR-2021 15:00               RM 359.80
TKD10002751            KD10002751             Gastroenteritis Care          24-APR-2021 18:00               RM 799.60
TKD10002750            KD10002750             Dental Treatment              24-APR-2021 17:00               RM 359.80
TKD10002748            KD10002748             Skin Care                     24-APR-2021 12:00               RM 479.90
TKD10002743            KD10002743             Gastroenteritis Care          22-APR-2021 16:00               RM 789.80
TKD10002741            KD10002741             Antibiotics Vaccination       21-APR-2021 17:00               RM 359.70
TKD10002740            KD10002740             Skin Care                     21-APR-2021 15:00               RM 354.90
TKD10002737            KD10002737             Gastroenteritis Care          20-APR-2021 14:00               RM 789.80
TKD10002736            KD10002736             Gastroenteritis Care          20-APR-2021 13:00               RM 789.80
TKD10002735            KD10002735             Dental Treatment              19-APR-2021 18:00               RM 359.80
TKD10002733            KD10002733             Skin Care                     19-APR-2021 15:00               RM 559.80
TKD10002732            KD10002732             Dental Treatment              19-APR-2021 14:00               RM 279.90
TKD10002728            KD10002728             Skin Care                     18-APR-2021 13:00               RM 479.90
TKD10002722            KD10002722             Dental Treatment              15-APR-2021 15:00               RM 279.90
TKD10002721            KD10002721             Antibiotics Vaccination       15-APR-2021 14:00               RM 359.70
TKD10002720            KD10002720             Antibiotics Vaccination       15-APR-2021 13:00               RM 359.70
TKD10002719            KD10002719             Skin Care                     14-APR-2021 15:00               RM 559.80
```



```
TKD10002717          KD10002717           Dental Treatment           13-APR-2021 17:00              RM 279.90
TKD10002716          KD10002716           Skin Care                  13-APR-2021 15:00              RM 434.80
TKD10002713          KD10002713           Skin Care                  12-APR-2021 18:00              RM 434.80
TKD10002709          KD10002709           Dental Treatment           11-APR-2021 13:00              RM 359.80
TKD10002708          KD10002708           Antibiotics Vaccination    10-APR-2021 16:00              RM 389.70
TKD10002703          KD10002703           Skin Care                  09-APR-2021 15:00              RM 559.80
TKD10002700          KD10002700           Pet Emergency Care         08-APR-2021 18:00              RM 679.60
TKD10002696          KD10002696           Antibiotics Vaccination    08-APR-2021 13:00              RM 359.70
TKD10002690          KD10002690           Antibiotics Vaccination    07-APR-2021 11:00              RM 359.70
TKD10002686          KD10002686           Dental Treatment           05-APR-2021 18:00              RM 279.90
TKD10002684          KD10002684           Antibiotics Vaccination    05-APR-2021 15:00              RM 389.70
TKD10002682          KD10002682           Antibiotics Vaccination    05-APR-2021 12:00              RM 449.60
TKD10002681          KD10002681           Pet Emergency Care         05-APR-2021 11:00              RM 619.70
TKD10002679          KD10002679           Skin Care                  03-APR-2021 17:00              RM 559.80
TKD10002677          KD10002677           Skin Care                  03-APR-2021 11:00              RM 559.80
TKD10002675          KD10002675           Skin Care                  02-APR-2021 17:00              RM 434.80
TKD10002671          KD10002671           Pet Emergency Care         02-APR-2021 11:00              RM 539.80
TKD10002665          KD10002665           Gastroenteritis Care       30-MAR-2021 15:00              RM 789.80
TKD10002663          KD10002663           Skin Care                  29-MAR-2021 16:00              RM 354.90
TKD10002658          KD10002658           Pet Emergency Care         28-MAR-2021 15:00              RM 679.60
TKD10002653          KD10002653           Dental Treatment           25-MAR-2021 18:00              RM 279.90
TKD10002650          KD10002650           Antibiotics Vaccination    25-MAR-2021 12:00              RM 299.80
TKD10002649          KD10002649           Gastroenteritis Care       24-MAR-2021 16:00              RM 939.70
TKD10002647          KD10002647           Gastroenteritis Care       24-MAR-2021 13:00              RM 999.60
TKD10002646          KD10002646           Antibiotics Vaccination    24-MAR-2021 11:00              RM 299.80
TKD10002642          KD10002642           Dental Treatment           23-MAR-2021 12:00              RM 359.80
TKD10002640          KD10002640           Gastroenteritis Care       22-MAR-2021 15:00              RM 789.80
TKD10002639          KD10002639           Antibiotics Vaccination    22-MAR-2021 14:00              RM 389.70
TKD10002638          KD10002638           Antibiotics Vaccination    22-MAR-2021 13:00              RM 449.60
TKD10002636          KD10002636           Skin Care                  20-MAR-2021 18:00              RM 354.90
TKD10002635          KD10002635           Gastroenteritis Care       20-MAR-2021 16:00              RM 589.80
TKD10002631          KD10002631           Antibiotics Vaccination    19-MAR-2021 12:00              RM 389.70
TKD10002627          KD10002627           Antibiotics Vaccination    17-MAR-2021 15:00              RM 359.70
TKD10002623          KD10002623           Skin Care                  16-MAR-2021 15:00              RM 434.80
TKD10002621          KD10002621           Pet Emergency Care         15-MAR-2021 16:00              RM 539.80
TKD10002620          KD10002620           Pet Emergency Care         15-MAR-2021 13:00              RM 619.70
TKD10002614          KD10002614           Dental Treatment           13-MAR-2021 15:00              RM 359.80
TKD10002611          KD10002611           Skin Care                  12-MAR-2021 12:00              RM 559.80
TKD10002609          KD10002609           Antibiotics Vaccination    11-MAR-2021 17:00              RM 449.60
TKD10002603          KD10002603           Gastroenteritis Care       10-MAR-2021 12:00              RM 589.80
TKD10002601          KD10002601           Gastroenteritis Care       09-MAR-2021 12:00              RM 739.70
TKD10002600          KD10002600           Gastroenteritis Care       09-MAR-2021 11:00              RM 999.60
TKD10002590          KD10002590           Skin Care                  06-MAR-2021 13:00              RM 479.90
TKD10002586          KD10002586           Skin Care                  04-MAR-2021 17:00              RM 559.80
TKD10002583          KD10002583           Gastroenteritis Care       03-MAR-2021 16:00              RM 789.80
TKD10002580          KD10002580           Gastroenteritis Care       03-MAR-2021 12:00              RM 999.60
TKD10002578          KD10002578           Dental Treatment           01-MAR-2021 18:00              RM 359.80
TKD10002571          KD10002571           Dental Treatment           27-FEB-2021 11:00              RM 279.90
TKD10002562          KD10002562           Dental Treatment           23-FEB-2021 16:00              RM 279.90
TKD10002560          KD10002560           Pet Emergency Care         23-FEB-2021 12:00              RM 539.80
TKD10002559          KD10002559           Gastroenteritis Care       22-FEB-2021 18:00              RM 849.70
TKD10002558          KD10002558           Gastroenteritis Care       22-FEB-2021 13:00              RM 849.70
TKD10002550          KD10002550           Skin Care                  20-FEB-2021 11:00              RM 354.90
TKD10002548          KD10002548           Gastroenteritis Care       19-FEB-2021 12:00              RM 849.70
TKD10002547          KD10002547           Dental Treatment           18-FEB-2021 16:00              RM 279.90
TKD10002546          KD10002546           Dental Treatment           18-FEB-2021 15:00              RM 279.90
TKD10002542          KD10002542           Pet Emergency Care         17-FEB-2021 16:00              RM 599.70
TKD10002533          KD10002533           Antibiotics Vaccination    15-FEB-2021 18:00              RM 389.70
TKD10002531          KD10002531           Skin Care                  15-FEB-2021 12:00              RM 354.90
TKD10002529          KD10002529           Skin Care                  14-FEB-2021 14:00              RM 479.90
TKD10002524          KD10002524           Pet Emergency Care         12-FEB-2021 18:00              RM 599.70
TKD10002522          KD10002522           Antibiotics Vaccination    12-FEB-2021 15:00              RM 449.60
TKD10002518          KD10002518           Gastroenteritis Care       11-FEB-2021 13:00              RM 789.80
TKD10002516          KD10002516           Dental Treatment           10-FEB-2021 18:00              RM 359.80
TKD10002515          KD10002515           Skin Care                  10-FEB-2021 16:00              RM 479.90
TKD10002513          KD10002513           Dental Treatment           09-FEB-2021 16:00              RM 279.90
```



```
TKD10002509          KD10002509          Antibiotics Vaccination      08-FEB-2021 18:00            RM 449.60
TKD10002506          KD10002506          Gastroenteritis Care         07-FEB-2021 16:00            RM 999.60
TKD10002499          KD10002499          Antibiotics Vaccination      06-FEB-2021 11:00            RM 449.60
TKD10002495          KD10002495          Skin Care                    04-FEB-2021 18:00            RM 434.80
TKD10002492          KD10002492          Pet Emergency Care           03-FEB-2021 17:00            RM 619.70
TKD10002488          KD10002488          Dental Treatment             03-FEB-2021 12:00            RM 279.90
TKD10002486          KD10002486          Gastroenteritis Care         01-FEB-2021 15:00            RM 799.60
TKD10002482          KD10002482          Dental Treatment             01-FEB-2021 11:00            RM 279.90
TKD10002478          KD10002478          Dental Treatment             30-JAN-2021 18:00            RM 359.80
TKD10002477          KD10002477          Pet Emergency Care           30-JAN-2021 15:00            RM 679.60
TKD10002476          KD10002476          Skin Care                    30-JAN-2021 12:00            RM 434.80
TKD10002474          KD10002474          Skin Care                    29-JAN-2021 15:00            RM 479.90
TKD10002473          KD10002473          Skin Care                    28-JAN-2021 18:00            RM 479.90
TKD10002467          KD10002467          Antibiotics Vaccination      27-JAN-2021 15:00            RM 299.80
TKD10002466          KD10002466          Skin Care                    26-JAN-2021 18:00            RM 434.80
TKD10002463          KD10002463          Skin Care                    26-JAN-2021 12:00            RM 559.80
TKD10002462          KD10002462          Pet Emergency Care           25-JAN-2021 18:00            RM 599.70
TKD10002460          KD10002460          Pet Emergency Care           25-JAN-2021 14:00            RM 539.80
TKD10002455          KD10002455          Pet Emergency Care           23-JAN-2021 13:00            RM 539.80
TKD10002453          KD10002453          Pet Emergency Care           22-JAN-2021 18:00            RM 599.70
TKD10002452          KD10002452          Gastroenteritis Care         22-JAN-2021 15:00            RM 999.60
TKD10002450          KD10002450          Gastroenteritis Care         22-JAN-2021 12:00            RM 649.70
TKD10002445          KD10002445          Pet Emergency Care           19-JAN-2021 17:00            RM 619.70
TKD10002443          KD10002443          Gastroenteritis Care         19-JAN-2021 13:00            RM 999.60
TKD10002440          KD10002440          Dental Treatment             17-JAN-2021 18:00            RM 359.80
TKD10002435          KD10002435          Antibiotics Vaccination      16-JAN-2021 14:00            RM 389.70
TKD10002432          KD10002432          Dental Treatment             15-JAN-2021 18:00            RM 279.90
TKD10002431          KD10002431          Skin Care                    15-JAN-2021 15:00            RM 434.80
TKD10002426          KD10002426          Pet Emergency Care           14-JAN-2021 17:00            RM 599.70
TKD10002424          KD10002424          Antibiotics Vaccination      14-JAN-2021 12:00            RM 389.70
TKD10002421          KD10002421          Pet Emergency Care           12-JAN-2021 16:00            RM 539.80
TKD10002420          KD10002420          Pet Emergency Care           12-JAN-2021 14:00            RM 679.60
TKD10002416          KD10002416          Gastroenteritis Care         10-JAN-2021 17:00            RM 849.70
TKD10002415          KD10002415          Gastroenteritis Care         10-JAN-2021 15:00            RM 649.70
TKD10002411          KD10002411          Pet Emergency Care           09-JAN-2021 18:00            RM 599.70
TKD10002406          KD10002406          Pet Emergency Care           07-JAN-2021 17:00            RM 599.70
TKD10002405          KD10002405          Antibiotics Vaccination      07-JAN-2021 16:00            RM 389.70
TKD10002396          KD10002396          Pet Emergency Care           05-JAN-2021 11:00            RM 539.80
TKD10002394          KD10002394          Gastroenteritis Care         04-JAN-2021 13:00            RM 739.70
=========================================================================================================
Total Treatment Done : 158                                      Total Amount of Transaction : RM 83,042.10
=========================================================================================================
```

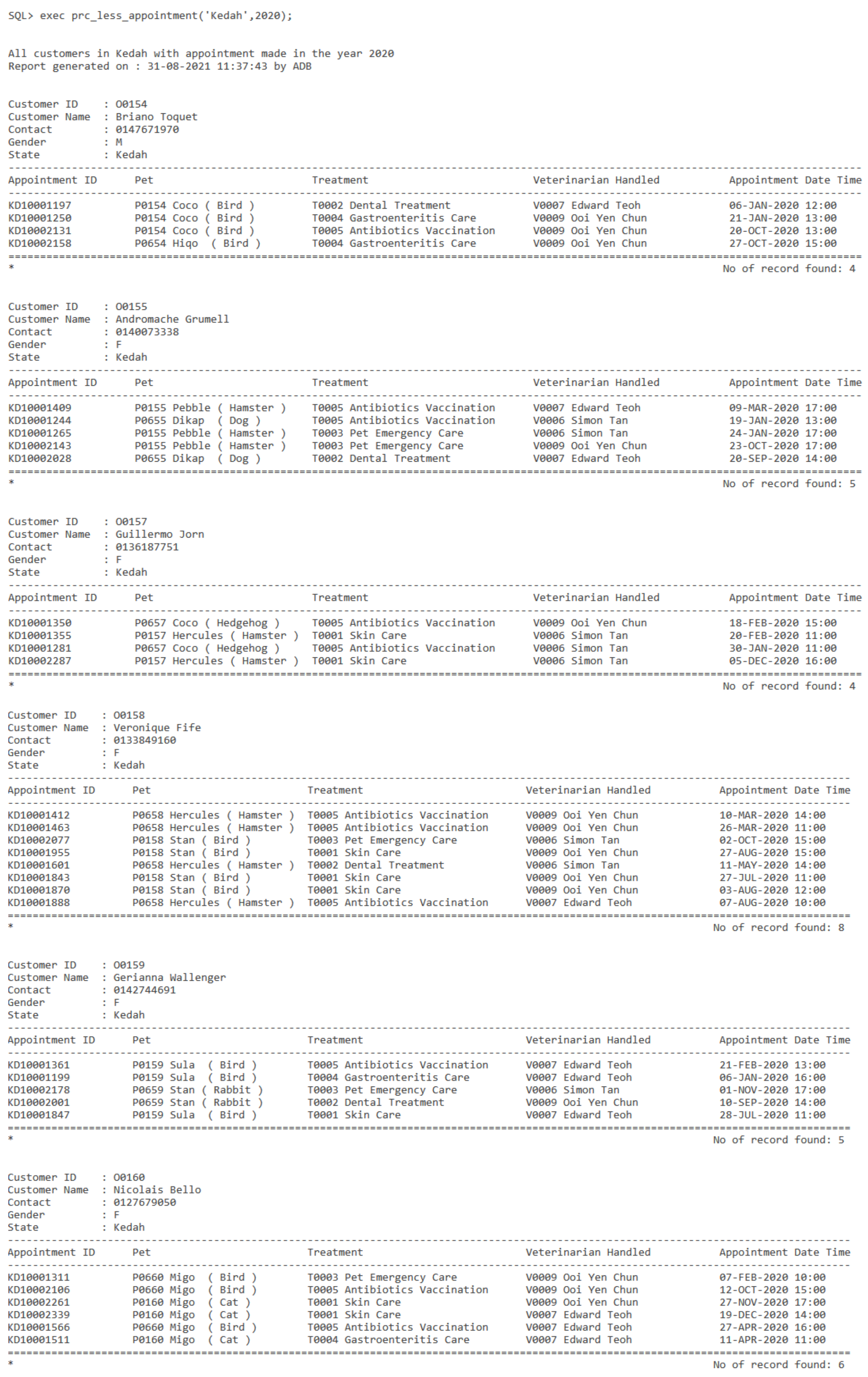
4.1.9 Report 2: Detail report of Customer list in specific state with appointment made in a year

Purpose: The purpose of this report is to list all customers with all of their appointments made in a specific state, in a selected year along with their details. It will provide in detail every appointment made by the customers in a year. Organizations in the clinic can view all the appointments made by every customer to see their potential loyal customers.

```
CREATE OR REPLACE PROCEDURE prc_less_appointment(IN_state IN
VARCHAR2,IN_year IN NUMBER) AS

   counter       NUMBER;
   record_count  NUMBER;
   e_norecord    EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_norecord,-20062);

   CURSOR cust_cursor IS
      SELECT owner_id, owner_name, owner_gender,
owner_contact, state
                                                FROM petowner
                                     WHERE state = IN_state;

   CURSOR app_cursor IS
      SELECT t.owner_id, a.appointment_id, a.pet_id,
p.pet_name, pt.type_name, a.treatment_id, tr.treatment_type,
a.vet_id, v.vet_name, a.appointment_dateTime
```

```
             FROM appointment a, transaction t, treatment tr,
veterinarian v, pet p, pettype pt
                  WHERE t.appointment_id=a.appointment_id AND
a.treatment_id=tr.treatment_id AND a.vet_id=v.vet_id AND
a.pet_id=p.pet_id AND p.type_id=pt.type_id AND EXTRACT(YEAR
FROM a.appointment_dateTime) = IN_year;

BEGIN
   record_count := 0;

   SELECT COUNT(*) INTO record_count
   FROM transaction t, appointment a, branch b
   WHERE t.appointment_id=a.appointment_id AND EXTRACT(YEAR
FROM a.appointment_dateTime) = IN_year AND b.state LIKE
IN_state;

   IF record_count = 0 THEN
     RAISE_APPLICATION_ERROR(-20062,'No record found');
   END IF;

   DBMS_OUTPUT.PUT_LINE (chr(10));
   DBMS_OUTPUT.PUT_LINE ('All customers in '||IN_state||'
with appointment made in the year '||IN_year);
   DBMS_OUTPUT.PUT_LINE('Report generated on : ' ||
TO_CHAR(CURRENT_DATE, 'DD-MM-YYYY HH:MI:SS') || ' by ' ||
USER);
   DBMS_OUTPUT.PUT_LINE (chr(10));

   FOR cust IN cust_cursor LOOP
      counter := 0;
      DBMS_OUTPUT.PUT_LINE ('Customer ID    :
'||cust.owner_ID);
      DBMS_OUTPUT.PUT_LINE ('Customer Name  :
'||cust.owner_name);
                     DBMS_OUTPUT.PUT_LINE ('Contact        :
'||cust.owner_contact);
                     DBMS_OUTPUT.PUT_LINE ('Gender         :
'||cust.owner_gender);
      DBMS_OUTPUT.PUT_LINE ('State          : '||cust.state);

      DBMS_OUTPUT.PUT_LINE(LPAD('-', 135, '-'));
      DBMS_OUTPUT.PUT_LINE(RPAD('Appointment ID', 20, ' ') ||
RPAD('Pet', 28, ' ') || RPAD('Treatment', 35, ' ')||
RPAD('Veterinarian Handled', 31, ' ') || RPAD('Appointment
Date Time', 25, ' '));
      DBMS_OUTPUT.PUT_LINE(LPAD('-', 135, '-'));

                                 FOR app IN app_cursor LOOP
                       IF app.owner_ID = cust.owner_ID THEN
          DBMS_OUTPUT.PUT_LINE(RPAD(app.appointment_id, 20, '
') || RPAD((app.pet_id||' '||app.pet_name||' (
'||app.type_name||')'), 28, ' ') || RPAD((app.treatment_id||'
'||app.treatment_type), 35, ' ')|| RPAD((app.vet_id||'
'||app.vet_name), 31, ' ') || RPAD(app.appointment_dateTime,
20, ' '));




                                      counter := counter + 1;

                                                      END IF;

                                                    END LOOP;

                    DBMS_OUTPUT.PUT_LINE(LPAD('=', 135, '='));
        DBMS_OUTPUT.PUT_LINE(RPAD('*',113,' ')||'No of record
found: '||counter);
                             DBMS_OUTPUT.PUT_LINE(CHR(10));

   END LOOP;

 EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE ('No Veterinarian found');
   WHEN e_norecord THEN

DBMS_OUTPUT.PUT_LINE('-----------------------------------------
------------');
      DBMS_OUTPUT.PUT_LINE('No record found for state
'||IN_state||' in year '||IN_year||'.');

DBMS_OUTPUT.PUT_LINE('-----------------------------------------
------------');
      DBMS_OUTPUT.PUT_LINE(SQLERRM);

END;
/

exec prc_less_appointment('Kedah',2020);
```

Sample Output:



```
SQL> exec prc_less_appointment('Kedah',2020);


All customers in Kedah with appointment made in the year 2020
Report generated on : 31-08-2021 11:37:43 by ADB


Customer ID    : O0154
Customer Name  : Briano Toquet
Contact        : 0147671970
Gender         : M
State          : Kedah
---------------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                         Treatment                          Veterinarian Handled           Appointment Date Time
---------------------------------------------------------------------------------------------------------------------------------------
KD10001197          P0154 Coco ( Bird )         T0002 Dental Treatment             V0007 Edward Teoh              06-JAN-2020 12:00
KD10001250          P0154 Coco ( Bird )         T0004 Gastroenteritis Care         V0009 Ooi Yen Chun             21-JAN-2020 13:00
KD10002131          P0154 Coco ( Bird )         T0005 Antibiotics Vaccination      V0009 Ooi Yen Chun             20-OCT-2020 13:00
KD10002158          P0654 Hiqo  ( Bird )        T0004 Gastroenteritis Care         V0009 Ooi Yen Chun             27-OCT-2020 15:00
=======================================================================================================================================
*                                                                                                                No of record found: 4


Customer ID    : O0155
Customer Name  : Andromache Grumell
Contact        : 0140073338
Gender         : F
State          : Kedah
---------------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                         Treatment                          Veterinarian Handled           Appointment Date Time
---------------------------------------------------------------------------------------------------------------------------------------
KD10001409          P0155 Pebble ( Hamster )    T0005 Antibiotics Vaccination      V0007 Edward Teoh              09-MAR-2020 17:00
KD10001244          P0655 Dikap  ( Dog )        T0005 Antibiotics Vaccination      V0006 Simon Tan                19-JAN-2020 13:00
KD10001265          P0155 Pebble ( Hamster )    T0003 Pet Emergency Care           V0006 Simon Tan                24-JAN-2020 17:00
KD10002143          P0155 Pebble ( Hamster )    T0003 Pet Emergency Care           V0009 Ooi Yen Chun             23-OCT-2020 17:00
KD10002028          P0655 Dikap  ( Dog )        T0002 Dental Treatment             V0007 Edward Teoh              20-SEP-2020 14:00
=======================================================================================================================================
*                                                                                                                No of record found: 5


Customer ID    : O0157
Customer Name  : Guillermo Jorn
Contact        : 0136187751
Gender         : F
State          : Kedah
---------------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                         Treatment                          Veterinarian Handled           Appointment Date Time
---------------------------------------------------------------------------------------------------------------------------------------
KD10001350          P0657 Coco ( Hedgehog )     T0005 Antibiotics Vaccination      V0009 Ooi Yen Chun             18-FEB-2020 15:00
KD10001355          P0157 Hercules ( Hamster )  T0001 Skin Care                    V0006 Simon Tan                20-FEB-2020 11:00
KD10001281          P0657 Coco ( Hedgehog )     T0005 Antibiotics Vaccination      V0006 Simon Tan                30-JAN-2020 11:00
KD10002287          P0157 Hercules ( Hamster )  T0001 Skin Care                    V0006 Simon Tan                05-DEC-2020 16:00
=======================================================================================================================================
*                                                                                                                No of record found: 4

Customer ID    : O0158
Customer Name  : Veronique Fife
Contact        : 0133849160
Gender         : F
State          : Kedah
---------------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                         Treatment                          Veterinarian Handled           Appointment Date Time
---------------------------------------------------------------------------------------------------------------------------------------
KD10001412          P0658 Hercules ( Hamster )  T0005 Antibiotics Vaccination      V0009 Ooi Yen Chun             10-MAR-2020 14:00
KD10001463          P0658 Hercules ( Hamster )  T0005 Antibiotics Vaccination      V0009 Ooi Yen Chun             26-MAR-2020 11:00
KD10002077          P0158 Stan ( Bird )         T0003 Pet Emergency Care           V0006 Simon Tan                02-OCT-2020 15:00
KD10001955          P0158 Stan ( Bird )         T0001 Skin Care                    V0009 Ooi Yen Chun             27-AUG-2020 15:00
KD10001601          P0658 Hercules ( Hamster )  T0002 Dental Treatment             V0006 Simon Tan                11-MAY-2020 14:00
KD10001843          P0158 Stan ( Bird )         T0001 Skin Care                    V0009 Ooi Yen Chun             27-JUL-2020 11:00
KD10001870          P0158 Stan ( Bird )         T0001 Skin Care                    V0009 Ooi Yen Chun             03-AUG-2020 12:00
KD10001888          P0658 Hercules ( Hamster )  T0005 Antibiotics Vaccination      V0007 Edward Teoh              07-AUG-2020 10:00
=======================================================================================================================================
*                                                                                                                No of record found: 8


Customer ID    : O0159
Customer Name  : Gerianna Wallenger
Contact        : 0142744691
Gender         : F
State          : Kedah
---------------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                         Treatment                          Veterinarian Handled           Appointment Date Time
---------------------------------------------------------------------------------------------------------------------------------------
KD10001361          P0159 Sula  ( Bird )        T0005 Antibiotics Vaccination      V0007 Edward Teoh              21-FEB-2020 13:00
KD10001199          P0159 Sula  ( Bird )        T0004 Gastroenteritis Care         V0007 Edward Teoh              06-JAN-2020 16:00
KD10002178          P0659 Stan ( Rabbit )       T0003 Pet Emergency Care           V0006 Simon Tan                01-NOV-2020 17:00
KD10002001          P0659 Stan ( Rabbit )       T0002 Dental Treatment             V0009 Ooi Yen Chun             10-SEP-2020 14:00
KD10001847          P0159 Sula  ( Bird )        T0001 Skin Care                    V0007 Edward Teoh              28-JUL-2020 11:00
=======================================================================================================================================
*                                                                                                                No of record found: 5


Customer ID    : O0160
Customer Name  : Nicolais Bello
Contact        : 0127679050
Gender         : F
State          : Kedah
---------------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                         Treatment                          Veterinarian Handled           Appointment Date Time
---------------------------------------------------------------------------------------------------------------------------------------
KD10001311          P0660 Migo  ( Bird )        T0003 Pet Emergency Care           V0009 Ooi Yen Chun             07-FEB-2020 10:00
KD10002106          P0660 Migo  ( Bird )        T0005 Antibiotics Vaccination      V0009 Ooi Yen Chun             12-OCT-2020 15:00
KD10002261          P0160 Migo  ( Cat )         T0001 Skin Care                    V0009 Ooi Yen Chun             27-NOV-2020 17:00
KD10002339          P0160 Migo  ( Cat )         T0001 Skin Care                    V0007 Edward Teoh              19-DEC-2020 14:00
KD10001566          P0660 Migo  ( Bird )        T0005 Antibiotics Vaccination      V0007 Edward Teoh              27-APR-2020 16:00
KD10001511          P0160 Migo  ( Cat )         T0004 Gastroenteritis Care         V0007 Edward Teoh              11-APR-2020 11:00
=======================================================================================================================================
*                                                                                                                No of record found: 6
```

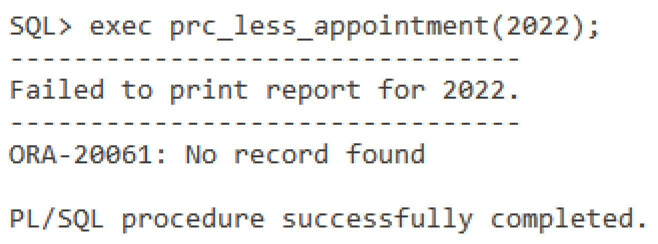
```
Customer ID    : O0164
Customer Name  : Ingrim Krienke
Contact        : 0183040252
Gender         : M
State          : Kedah
--------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                          Treatment                         Veterinarian Handled           Appointment Date Time
--------------------------------------------------------------------------------------------------------------------------------
KD10002100          P0164 Max ( Rabbit )         T0004 Gastroenteritis Care        V0006 Simon Tan                10-OCT-2020 16:00
KD10002266          P0664 Kayle ( Bird )         T0003 Pet Emergency Care          V0006 Simon Tan                29-NOV-2020 11:00
KD10001972          P0664 Kayle ( Bird )         T0004 Gastroenteritis Care        V0009 Ooi Yen Chun             01-SEP-2020 12:00
KD10001935          P0664 Kayle ( Bird )         T0001 Skin Care                   V0007 Edward Teoh              19-AUG-2020 13:00
================================================================================================================================
*                                                                                                          No of record found: 4


Customer ID    : O0165
Customer Name  : Rolf Brason
Contact        : 0173330767
Gender         : F
State          : Kedah
--------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                          Treatment                         Veterinarian Handled           Appointment Date Time
--------------------------------------------------------------------------------------------------------------------------------
KD10001376          P0665 Nosey ( Dog )          T0004 Gastroenteritis Care        V0006 Simon Tan                25-FEB-2020 13:00
KD10002080          P0665 Nosey ( Dog )          T0001 Skin Care                   V0009 Ooi Yen Chun             03-OCT-2020 14:00
KD10002104          P0165 Zoey ( Dog )           T0004 Gastroenteritis Care        V0007 Edward Teoh              11-OCT-2020 14:00
KD10002185          P0165 Zoey ( Dog )           T0004 Gastroenteritis Care        V0009 Ooi Yen Chun             03-NOV-2020 15:00
KD10002026          P0165 Zoey ( Dog )           T0002 Dental Treatment            V0009 Ooi Yen Chun             19-SEP-2020 16:00
KD10001942          P0165 Zoey ( Dog )           T0005 Antibiotics Vaccination     V0009 Ooi Yen Chun             21-AUG-2020 15:00
================================================================================================================================
*                                                                                                          No of record found: 6


Customer ID    : O0169
Customer Name  : Bell Tanman
Contact        : 0191150579
Gender         : F
State          : Kedah
--------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                          Treatment                         Veterinarian Handled           Appointment Date Time
--------------------------------------------------------------------------------------------------------------------------------
KD10002181          P0169 Kiwi  ( Hamster )      T0003 Pet Emergency Care          V0006 Simon Tan                02-NOV-2020 16:00
KD10001649          P0169 Kiwi  ( Hamster )      T0005 Antibiotics Vaccination     V0009 Ooi Yen Chun             26-MAY-2020 17:00
KD10001742          P0169 Kiwi  ( Hamster )      T0002 Dental Treatment            V0007 Edward Teoh              25-JUN-2020 13:00
KD10001748          P0169 Kiwi  ( Hamster )      T0004 Gastroenteritis Care        V0007 Edward Teoh              26-JUN-2020 14:00
================================================================================================================================
*                                                                                                          No of record found: 4


Customer ID    : O0170
Customer Name  : Wernher Mallion
Contact        : 0166651345
Gender         : F
State          : Kedah
--------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                          Treatment                         Veterinarian Handled           Appointment Date Time
--------------------------------------------------------------------------------------------------------------------------------
KD10002053          P0670 Saga ( Hedgehog )      T0002 Dental Treatment            V0009 Ooi Yen Chun             27-SEP-2020 14:00
KD10002187          P0170 Jaws  ( Bird )         T0002 Dental Treatment            V0009 Ooi Yen Chun             04-NOV-2020 10:00
KD10001895          P0670 Saga ( Hedgehog )      T0001 Skin Care                   V0007 Edward Teoh              08-AUG-2020 14:00
================================================================================================================================
*                                                                                                          No of record found: 3

Customer ID    : O0246
Customer Name  : Roselia Rutigliano
Contact        : 0102862669
Gender         : F
State          : Kedah
--------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                          Treatment                         Veterinarian Handled           Appointment Date Time
--------------------------------------------------------------------------------------------------------------------------------
KD10001360          P0746 Tune ( Bird )          T0001 Skin Care                   V0007 Edward Teoh              21-FEB-2020 11:00
KD10001413          P0746 Tune ( Bird )          T0005 Antibiotics Vaccination     V0009 Ooi Yen Chun             10-MAR-2020 16:00
KD10001960          P0246 Suci  ( Cat )          T0003 Pet Emergency Care          V0006 Simon Tan                29-AUG-2020 12:00
KD10001600          P0246 Suci  ( Cat )          T0002 Dental Treatment            V0009 Ooi Yen Chun             11-MAY-2020 13:00
KD10001721          P0746 Tune ( Bird )          T0005 Antibiotics Vaccination     V0006 Simon Tan                19-JUN-2020 10:00
KD10001852          P0746 Tune ( Bird )          T0004 Gastroenteritis Care        V0007 Edward Teoh              30-JUL-2020 12:00
KD10001915          P0246 Suci  ( Cat )          T0003 Pet Emergency Care          V0009 Ooi Yen Chun             13-AUG-2020 12:00
KD10001502          P0746 Tune ( Bird )          T0005 Antibiotics Vaccination     V0009 Ooi Yen Chun             07-APR-2020 14:00
================================================================================================================================
*                                                                                                          No of record found: 8

Customer ID    : O0247
Customer Name  : Rodolfo Duerdin
Contact        : 0185529739
Gender         : F
State          : Kedah
--------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                          Treatment                         Veterinarian Handled           Appointment Date Time
--------------------------------------------------------------------------------------------------------------------------------
KD10001189          P0747 Jan ( Hamster )        T0001 Skin Care                   V0007 Edward Teoh              04-JAN-2020 11:00
================================================================================================================================
*                                                                                                          No of record found: 1

Customer ID    : O0248
Customer Name  : Bernie Giuron
Contact        : 0142712049
Gender         : F
State          : Kedah
--------------------------------------------------------------------------------------------------------------------------------
Appointment ID      Pet                          Treatment                         Veterinarian Handled           Appointment Date Time
--------------------------------------------------------------------------------------------------------------------------------
KD10001729          P0248 Huqi ( Bird )          T0005 Antibiotics Vaccination     V0007 Edward Teoh              20-JUN-2020 14:00
================================================================================================================================
*                                                                                                          No of record found: 1
```

4.1.10 Report 3: On-demand report of Customer list who do transaction for less than 3 times in a year

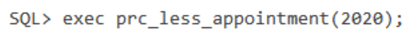
Purpose: The purpose of this report is to list out the customers who have less than 3 transactions done in a year. This can let the management to predict the churn customer by viewing this report which also includes the last transaction made by the customer. In this way, they can try to ask the listed customers for their feedback and satisfaction against the clinic to know why they will be churning.

```
CREATE OR REPLACE PROCEDURE prc_less_appointment(IN_year IN
NUMBER) AS

   counter       NUMBER;
   record_count  NUMBER;
   e_norecord    EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_norecord,-20061);

  CURSOR cust_trans IS
         SELECT po.owner_id, po.owner_name, po.owner_contact,
po.owner_gender, po.state, COUNT(t.transaction_dateTime) AS
TotalTransaction, MAX(t.transaction_dateTime) AS
LastTransaction
                              FROM petowner po, transaction t
           WHERE t.owner_id=po.owner_id AND EXTRACT(YEAR FROM
t.transaction_dateTime) = IN_year
       GROUP BY po.owner_id, po.owner_name, po.owner_contact,
po.owner_gender, po.state;

BEGIN
   counter := 0;
   record_count := 0;

   SELECT COUNT(*) INTO record_count
   FROM petowner po, transaction t
   WHERE t.owner_id=po.owner_id AND EXTRACT(YEAR FROM
t.transaction_dateTime) = IN_year;

   IF record_count = 0 THEN
     RAISE_APPLICATION_ERROR(-20061,'No record found');
   END IF;

   DBMS_OUTPUT.PUT_LINE (chr(10));
   DBMS_OUTPUT.PUT_LINE ('List of customer who do transaction
for less than 3 times in the year '||IN_year);
   DBMS_OUTPUT.PUT_LINE('Report generated on : ' ||
TO_CHAR(CURRENT_DATE, 'DD-MM-YYYY HH:MI:SS') || ' by ' ||
USER);
   DBMS_OUTPUT.PUT_LINE (chr(10));

   DBMS_OUTPUT.PUT_LINE(LPAD('-', 137, '-'));
   DBMS_OUTPUT.PUT_LINE(RPAD('Customer ID', 15, ' ') ||
RPAD('Customer Name', 26, ' ') || RPAD('Customer Contact',
20, ' ')|| RPAD('Gender', 10, ' ') || RPAD('State', 20, ' ')
|| RPAD('No of Transaction made', 25, ' ') || RPAD('Last
Transaction Date', 30, ' '));
   DBMS_OUTPUT.PUT_LINE(LPAD('-', 137, '-'));

   FOR cust_record IN cust_trans LOOP
      IF cust_record.totaltransaction < 3 THEN
     DBMS_OUTPUT.PUT_LINE(RPAD(cust_record.owner_id, 15, ' ')
|| RPAD(cust_record.owner_name, 26, ' ') ||
RPAD(cust_record.owner_contact, 20, ' ')||
RPAD(cust_record.owner_gender, 10, ' ') ||
RPAD(cust_record.state, 20, ' ') ||
RPAD(cust_record.totaltransaction, 25, ' ') ||
RPAD(cust_record.lasttransaction, 30, ' '));



                                   counter := counter + 1;
                                                   END IF;

   END LOOP;

   DBMS_OUTPUT.PUT_LINE(LPAD('=', 137, '='));
   DBMS_OUTPUT.PUT_LINE(RPAD('*',110,' ')||'Total No of
Customer: '||counter);
   DBMS_OUTPUT.PUT_LINE(LPAD('=', 137, '='));

 EXCEPTION
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE ('No Veterinarian found');
   WHEN e_norecord THEN

DBMS_OUTPUT.PUT_LINE('--------------------------------');
      DBMS_OUTPUT.PUT_LINE('Failed to print report for ' ||
IN_year || '.');

DBMS_OUTPUT.PUT_LINE('--------------------------------');
      DBMS_OUTPUT.PUT_LINE(SQLERRM);

END;
/

exec prc_less_appointment(2020);
```

Sample Output:



```
SQL> exec prc_less_appointment(2022);
--------------------------------
Failed to print report for 2022.
--------------------------------
ORA-20061: No record found

PL/SQL procedure successfully completed.
```



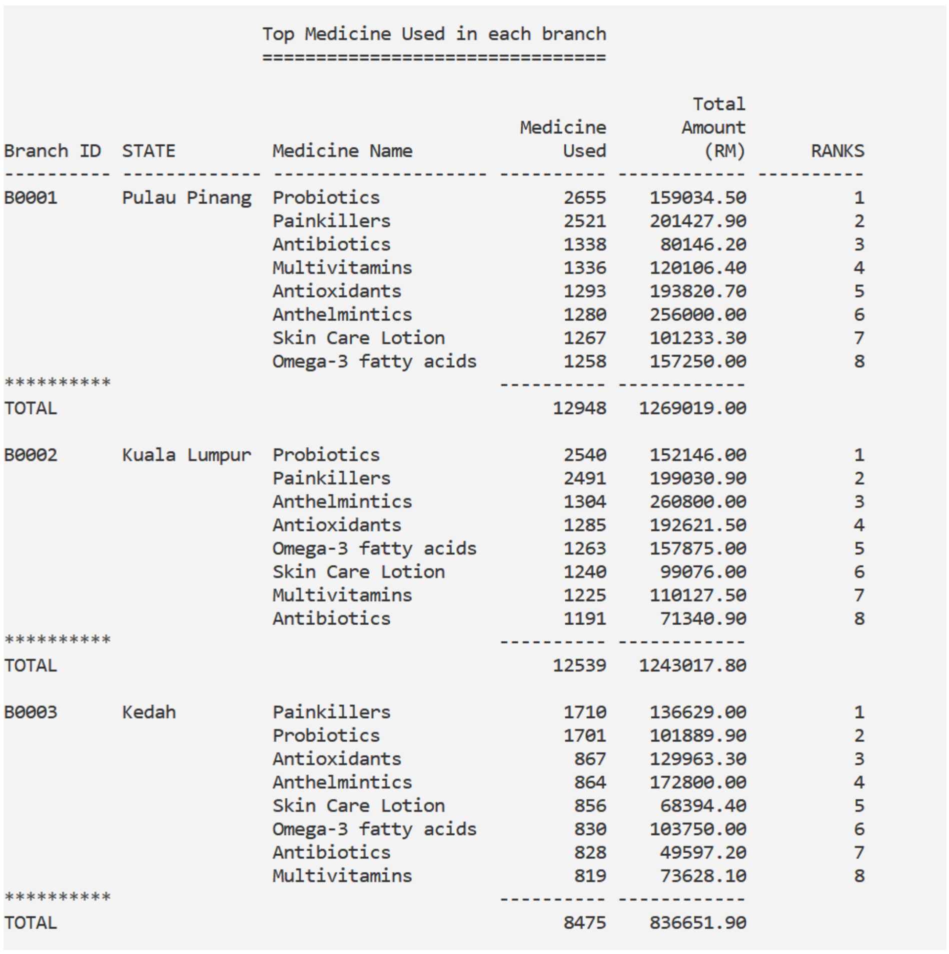
```
SQL> exec prc_less_appointment(2020);
```

4.2 (Tan Teoh Xin Ee)

4.2.1 Query 1:Top Medicine Used in each branch (Strategic)
Purpose: The purpose of this query is to analyze the most used medicine in each branch. Therefore, we are able to tackle each and every branch using this query. For example, this medicine has the highest use in this branch, therefore the medicine is in demand in this area which we will have to focus the growth and stock in more of that kind of medicine to that branch.

SQL statement:
```
clear break
clear compute
set linesize 80
set pagesize 100
break on state on branch_id skip 1
compute SUM Label TOTAL of quantity percentage amount on
branch_id
TTITLE ON
TTITLE CENTER 'Top Medicine Used in each branch' SKIP 1-
CENTER ================================= SKIP 2
column branch_id format a10
column branch_id heading 'Branch ID'
column state format a13
column medic_name format a20
column medic_name heading 'Medicine Name'
column quantity heading 'Medicine|Used'
column amount format 99999999.99
column amount heading 'Total|Amount|(RM)'

create or replace view medicalUsed As
select t.branch_id, d.medic_id, m.medic_name,
sum(d.line_qty) as quantity, sum(d.line_total) as amount
from transaction t, branch b, transactiondetail d,
medicalsupply m
where b.branch_id = t.branch_id
      and t.transaction_id = d.transaction_id
      and d.medic_id = m.medic_id
group by t.branch_id, d.medic_id,m.medic_name
order by t.branch_id,sum(d.line_qty) DESC;

select u.branch_id, b.state, u.medic_name, u.quantity,
u.amount,
RANK() over(partition by u.branch_id order by u.quantity
DESC) Ranks
from medicalused u, transaction t, branch b
where u.branch_id = t.branch_id
      and u.branch_id = b.branch_id
group by u.branch_id, b.state, u.medic_name, u.quantity,
u.amount
order by u.branch_id,u.quantity DESC;
```

Sample Output:



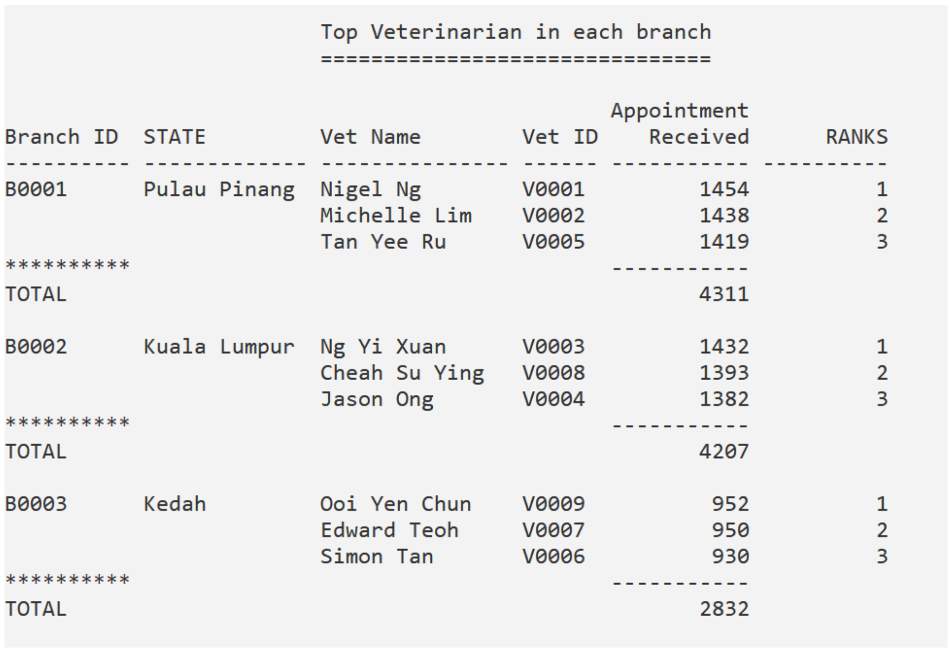
```
                        Top Medicine Used in each branch
                        ================================

                                                             Total
                                                Medicine    Amount
Branch ID  STATE         Medicine Name              Used      (RM)      RANKS
---------- ------------- -------------------- ---------- ------------ ----------
B0001      Pulau Pinang  Probiotics                 2655    159034.50          1
                         Painkillers                2521    201427.90          2
                         Antibiotics                1338     80146.20          3
                         Multivitamins              1336    120106.40          4
                         Antioxidants               1293    193820.70          5
                         Anthelmintics              1280    256000.00          6
                         Skin Care Lotion           1267    101233.30          7
                         Omega-3 fatty acids        1258    157250.00          8
**********                                    ---------- ------------
TOTAL                                              12948   1269019.00

B0002      Kuala Lumpur  Probiotics                 2540    152146.00          1
                         Painkillers                2491    199030.90          2
                         Anthelmintics              1304    260800.00          3
                         Antioxidants               1285    192621.50          4
                         Omega-3 fatty acids        1263    157875.00          5
                         Skin Care Lotion           1240     99076.00          6
                         Multivitamins              1225    110127.50          7
                         Antibiotics                1191     71340.90          8
**********                                    ---------- ------------
TOTAL                                              12539   1243017.80

B0003      Kedah         Painkillers                1710    136629.00          1
                         Probiotics                 1701    101889.90          2
                         Antioxidants                867    129963.30          3
                         Anthelmintics               864    172800.00          4
                         Skin Care Lotion            856     68394.40          5
                         Omega-3 fatty acids         830    103750.00          6
                         Antibiotics                 828     49597.20          7
                         Multivitamins               819     73628.10          8
**********                                    ---------- ------------
TOTAL                                               8475    836651.90
```

4.2.2 Query 2: Top Veterinarian in each branch (Tactical)
Purpose: The purpose of this query is to know the most appointments received by each veterinarian in every branch. Therefore, we can revise the staffing levels by looking at their performance. For example, the most appointed veterinarian can be promoted to senior veterinarian.

SQL statement:

```
clear break
clear compute
set linesize 80
set pagesize 100
break on state on branch_id skip 1
compute SUM Label TOTAL of noofapp on branch_id
TTITLE ON
TTITLE CENTER 'Top Veterinarian in each branch' SKIP 1-
CENTER =============================== SKIP 2
column branch_id format a10
column branch_id heading 'Branch ID'
column state format a13
column vet_id format a6
column vet_id heading 'Vet ID'
column vet_name format a15
column vet_name heading 'Vet Name'
column noofapp heading 'Appointment|Received'

create or replace view appointNum As
select count(appointment_id) as NoOfapp, vet_id
from appointment
group by vet_id
order by count(appointment_id) desc;

select b.branch_id, b.state, v.vet_name, a.vet_id,
a.noofapp,
RANK() over(partition by b.branch_id order by a.noofapp
DESC) RANKS
from branch b, veterinarian v, appointNum a
where a.vet_id = v.vet_id
      and v.branch_id = b.branch_id
group by b.branch_id, b.state, v.vet_name, a.vet_id,
a.noofapp
order by b.branch_id,a.noofapp DESC;
```

Sample Output:



```
                         Top Veterinarian in each branch
                         ===============================

                                                  Appointment
Branch ID  STATE         Vet Name        Vet ID      Received      RANKS
---------- ------------- --------------- ------ ----------- ----------
B0001      Pulau Pinang  Nigel Ng        V0001         1454          1
                         Michelle Lim    V0002         1438          2
                         Tan Yee Ru      V0005         1419          3
**********                                      -----------
TOTAL                                                  4311

B0002      Kuala Lumpur  Ng Yi Xuan      V0003         1432          1
                         Cheah Su Ying   V0008         1393          2
                         Jason Ong       V0004         1382          3
**********                                      -----------
TOTAL                                                  4207

B0003      Kedah         Ooi Yen Chun    V0009          952          1
                         Edward Teoh     V0007          950          2
                         Simon Tan       V0006          930          3
**********                                      -----------
TOTAL                                                  2832
```

4.2.3 Query 3: Late Supplier List (Operational)

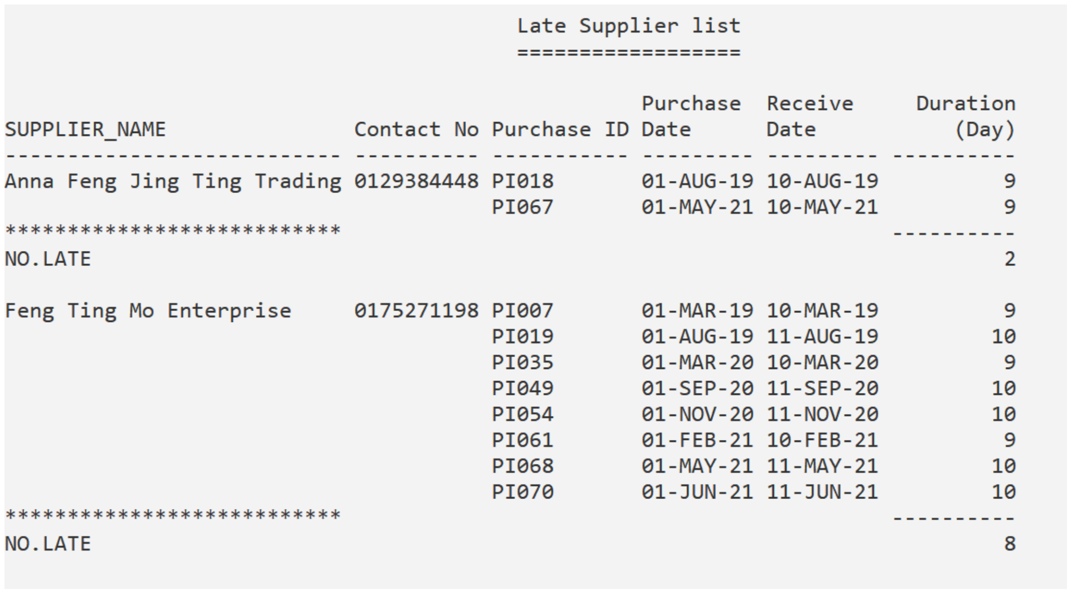
Purpose: The purpose of this query is to know the most frequent late supplier. By using this query, we are able to manage our stock such as the correct order/stock delay. This query is normally used by the low level staff in the shop.

SQL statement:

```
clear break
clear compute
set linesize 100
set pagesize 150
BREAK ON REPORT
```

```
break on supplier_contact on supplier_name skip 1
compute Count Label NO.LATE of duration on supplier_name
TTITLE ON
TTITLE CENTER 'Late Supplier list' SKIP 1-
CENTER ================== SKIP 2
column supplier_name format a27
column supplier_contact format a10
column supplier_contact heading 'Contact No'
column purchase_id format a11
column purchase_id heading 'Purchase ID'
column purchase_date format a9
column purchase_date heading 'Purchase|Date'
column receive_date format a9
column receive_date heading 'Receive|Date'
column duration heading 'Duration|(Day)'

create or replace view difdate as
select purchase_id, supplier_id, purchase_date,
receive_date,(receive_date-purchase_date) as duration
from purchaseTransaction
where receive_date-purchase_date>6
group by purchase_id,supplier_id, purchase_date,
receive_date
order by supplier_id;

select s.supplier_name,
s.supplier_contact,d.purchase_id, d.purchase_date,
d.receive_date,d.duration
from difdate d, supplier s
where d.supplier_id = s.supplier_id
group by s.supplier_name,
s.supplier_contact,d.purchase_id, d.purchase_date,
d.receive_date,d.duration
order by s.supplier_name;
```

Sample Output:



```
                                       Late Supplier list
                                       ==================

                                                 Purchase  Receive     Duration
SUPPLIER_NAME              Contact No Purchase ID Date      Date         (Day)
-------------------------- ---------- ----------- --------- --------- ----------
Anna Feng Jing Ting Trading 0129384448 PI018       01-AUG-19 10-AUG-19          9
                                      PI067       01-MAY-21 10-MAY-21          9
**************************                                            ----------
NO.LATE                                                                         2

Feng Ting Mo Enterprise    0175271198 PI007       01-MAR-19 10-MAR-19          9
                                      PI019       01-AUG-19 11-AUG-19         10
                                      PI035       01-MAR-20 10-MAR-20          9
                                      PI049       01-SEP-20 11-SEP-20         10
                                      PI054       01-NOV-20 11-NOV-20         10
                                      PI061       01-FEB-21 10-FEB-21          9
                                      PI068       01-MAY-21 11-MAY-21         10
                                      PI070       01-JUN-21 11-JUN-21         10
**************************                                            ----------
NO.LATE                                                                         8
```

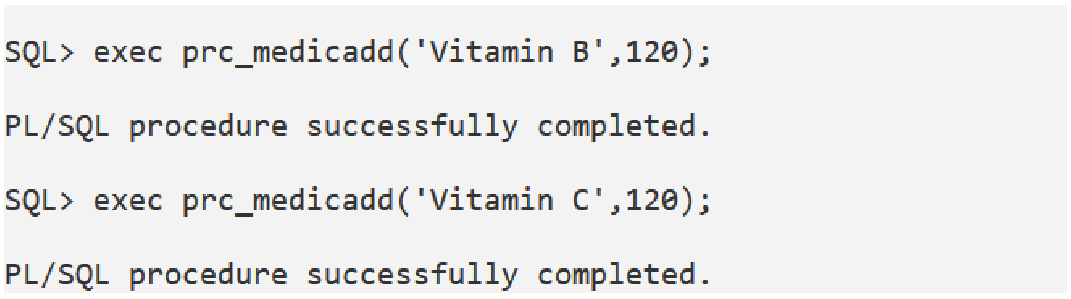
4.2.4 Procedure 1: Add Medical Supply
Purpose: The purpose of this procedure is to add new medicine into our database. If the user wishes to add new medicine, he/she can just call this procedure with two parameters(medicine name & price). Then the new record will auto be generated.

SQL statement:

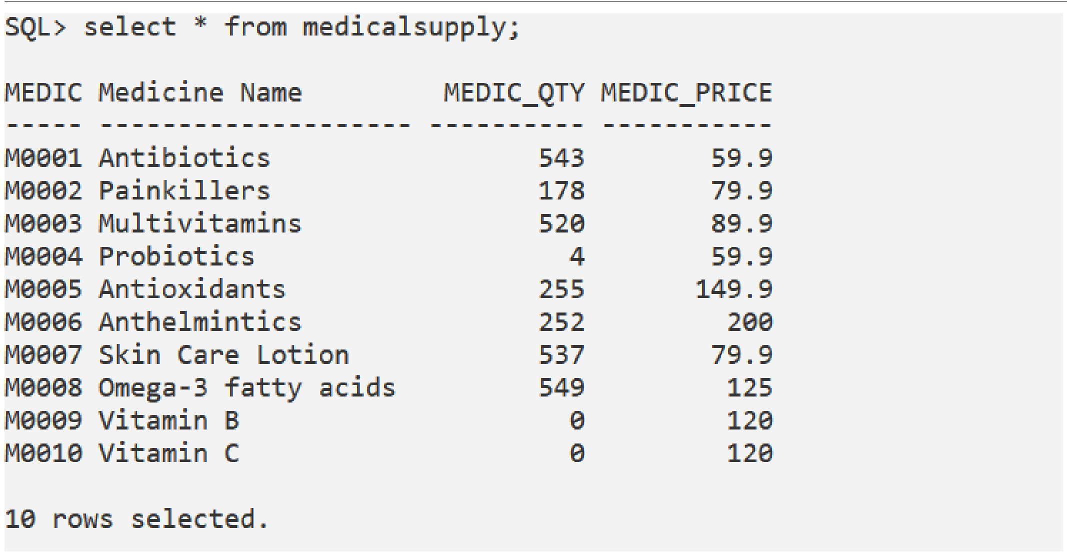
```
DROP SEQUENCE MEDICID;

CREATE SEQUENCE MEDICID
  MINVALUE 8
  MAXVALUE 9999
  START WITH 9
  INCREMENT BY 1;

CREATE OR REPLACE PROCEDURE PRC_MEDICADD (IN_MEDICNAME
IN VARCHAR, IN_MEDICPRICE IN NUMBER) As
  v_insertID char(5);
  NEXT NUMBER;

BEGIN
  NEXT:=MEDICID.NEXTVAL;
  IF (NEXT<10) THEN
   V_INSERTID := TO_CHAR('M000'||NEXT);
  ELSIF (NEXT<100) THEN
   V_INSERTID := TO_CHAR('M00'||NEXT);
  ELSIF (NEXT<1000) THEN
   V_INSERTID := TO_CHAR('M0'||NEXT);
  ELSIF (NEXT<10000) THEN
   V_INSERTID := TO_CHAR('M'||NEXT);
  END IF;

  INSERT INTO MEDICALSUPPLY
VALUES(V_INSERTID,IN_MEDICNAME,0,IN_MEDICPRICE);
```

```
END;
/
```

Sample Output:



```
SQL> exec prc_medicadd('Vitamin B',120);

PL/SQL procedure successfully completed.

SQL> exec prc_medicadd('Vitamin C',120);

PL/SQL procedure successfully completed.
```



```
SQL> select * from medicalsupply;

MEDIC Medicine Name         MEDIC_QTY MEDIC_PRICE
----- -------------------- ---------- -----------
M0001 Antibiotics                 543        59.9
M0002 Painkillers                 178        79.9
M0003 Multivitamins               520        89.9
M0004 Probiotics                    4        59.9
M0005 Antioxidants                255       149.9
M0006 Anthelmintics               252         200
M0007 Skin Care Lotion            537        79.9
M0008 Omega-3 fatty acids         549         125
M0009 Vitamin B                     0         120
M0010 Vitamin C                     0         120

10 rows selected.
```

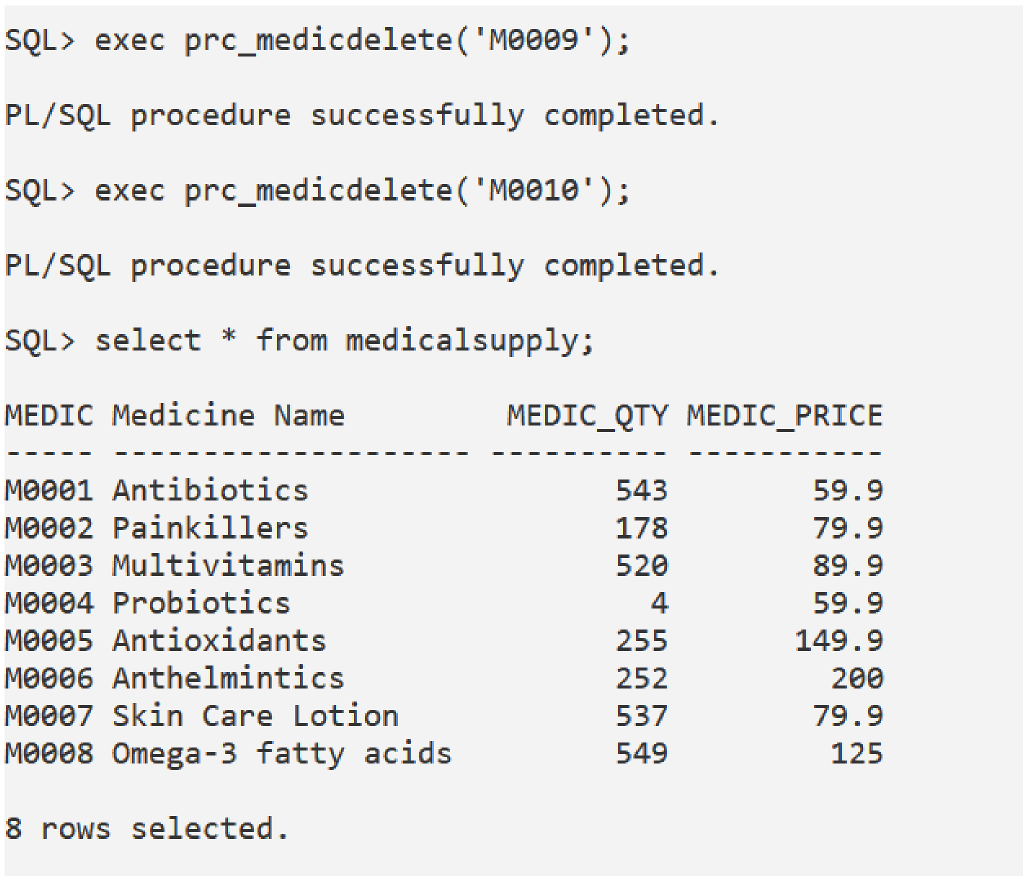
4.2.5 Procedure 2: Delete Medical Supply

Purpose: The purpose of this procedure is to delete unwanted medicine from our database. If the user wishes to delete the medicine, he/she can just call this procedure with one parameter(medicine id). Then the specific record will be deleted permanently.

SQL statement:

```
CREATE OR REPLACE PROCEDURE PRC_MEDICDELETE(IN_MEDICID
IN CHAR) AS
BEGIN
   DELETE FROM medicalsupply
   WHERE  medic_id = IN_MEDICID;

Exception
   WHEN NO_DATA_FOUND THEN
      DBMS_OUTPUT.PUT_LINE('NO SUCH MEDICINE!!!');
END;
/
```

Sample Output:



```
SQL> exec prc_medicdelete('M0009');

PL/SQL procedure successfully completed.

SQL> exec prc_medicdelete('M0010');

PL/SQL procedure successfully completed.

SQL> select * from medicalsupply;

MEDIC Medicine Name          MEDIC_QTY MEDIC_PRICE
----- -------------------- ---------- -----------
M0001 Antibiotics                 543        59.9
M0002 Painkillers                 178        79.9
M0003 Multivitamins               520        89.9
M0004 Probiotics                    4        59.9
M0005 Antioxidants                255       149.9
M0006 Anthelmintics               252         200
M0007 Skin Care Lotion            537        79.9
M0008 Omega-3 fatty acids         549         125

8 rows selected.
```

4.2.6 Trigger 1: Add Stock Quantity (Add after purchasing stock)
Purpose: The purpose of this trigger is used to trigger after purchasing stock. For example, it will automatically update and add specific quantities of stock, after the company has purchased stock.

SQL statement:
```
CREATE OR REPLACE TRIGGER TRG_Add_Stock_Quantity
  After Insert ON PurchaseItem
  FOR EACH ROW
 BEGIN
  Update MedicalSupply
   SET medic_qty = medic_qty + :new.purchase_qty
   where medic_id = :new.medic_id;
END;
/
```

4.2.7 Trigger 2: Minus Stock Quantity (Minus after adding transaction detail)
Purpose: The purpose of this trigger is used to trigger after transaction detail. For example, it will automatically update and delete specific quantities of stock, after every transaction is made by the customer.

SQL statement:
```
CREATE OR REPLACE TRIGGER TRG_Minus_Stock_Quantity
  After Insert ON TransactionDetail
  FOR EACH ROW
```

```
 BEGIN
  Update MedicalSupply
   SET medic_qty = medic_qty - :new.line_qty
   where medic_id = :new.medic_id;
END;
/
```

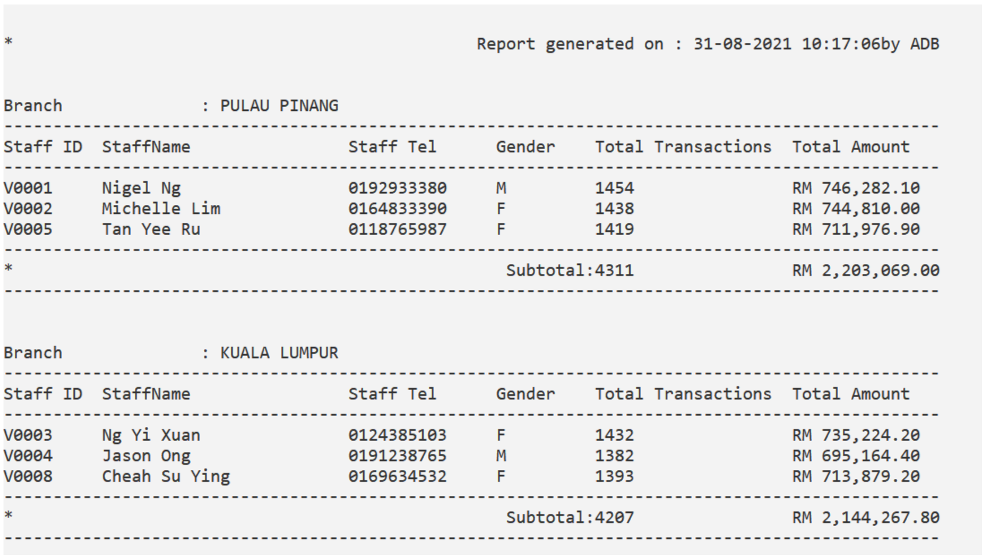
4.2.8 Report 1: Detail report of Veterinarian Performance
Purpose: The purpose of this report is to know every veterinarian's performance in their branch such as how many transactions they have made and how much profit they bring toward the company. At the end, we are also able to know the grand total made by these three branches.

SQL statement:

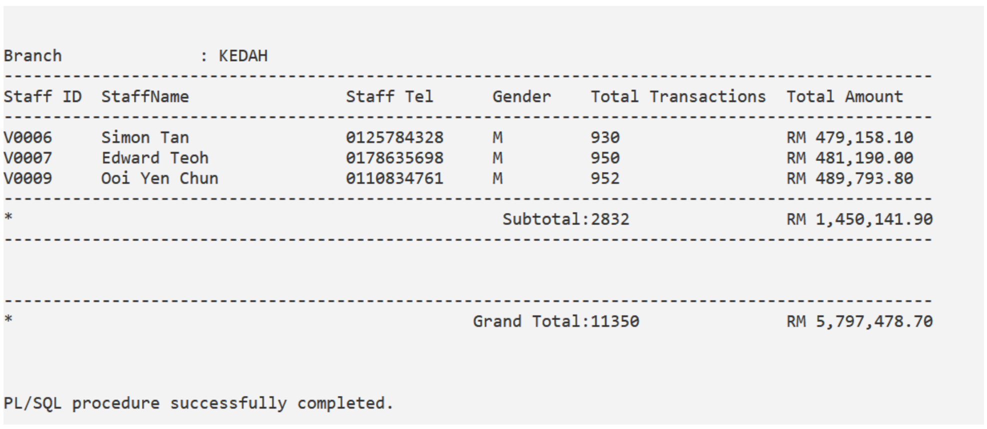
```
CREATE OR REPLACE PROCEDURE PRC_VET_PERFORMANCE IS

CURSOR BRANCH_CURSOR IS
 SELECT DISTINCT b.state
 FROM veterinarian v, branch b
 where v.branch_id = b.branch_id
 ORDER BY b.state desc;

 cursor vet_cursor (branches in char)is
 select v.vet_id, b.state, v.vet_name, v.vet_contact,
v.vet_gender, count(t.transaction_id) as transaction,
sum(t.total_amount)as amount
 from veterinarian v, appointment a, transaction t,
branch b
 where v.vet_id= a.vet_id
       and v.branch_id = b.branch_id
       and b.state = branches
       and a.appointment_id = t.appointment_id
  group by v.vet_id, b.state, v.vet_name, v.vet_contact,
v.vet_gender
  order by v.vet_id;
 v_totalAmt NUMBER(17,2) := 0;
 v_totalTrans NUMBER(10) := 0;
 v_grandTotal NUMBER(17,2) := 0;
 v_grandTrans NUMBER(10) := 0;

BEGIN

DBMS_OUTPUT.PUT_LINE(chr(10));
DBMS_OUTPUT.PUT_LINE(RPAD('*', 48, ' ') || 'Report
generated on : ' || TO_CHAR(CURRENT_DATE, 'DD-MM-YYYY
HH:MI:SS') || 'by ' || USER);
DBMS_OUTPUT.PUT_LINE(chr(10));

FOR branches IN BRANCH_CURSOR LOOP
DBMS_OUTPUT.PUT_LINE(RPAD('Branch', 20, ' ') || ': '
||RPAD(UPPER(branches.state), 60, ' '));
```

```
DBMS_OUTPUT.PUT_LINE(LPAD('-', 95, '-'));
DBMS_OUTPUT.PUT_LINE(RPAD('Staff ID', 10, ' ') ||
RPAD('StaffName', 25, ' ') || RPAD('Staff Tel', 15, ' ')
|| RPAD('Gender', 10, ' ')|| RPAD('Total Transactions',
20, ' ') || RPAD('Total Amount', 17, ' '));
DBMS_OUTPUT.PUT_LINE(LPAD('-', 95, '-'));
v_totalTrans := 0;
v_totalAmt := 0;

FOR vet_rec IN vet_CURSOR(branches.state) LOOP
DBMS_OUTPUT.PUT_LINE(RPAD(vet_rec.vet_id, 10, ' ')
||RPAD(vet_rec.vet_name, 25, ' ') ||
RPAD(vet_rec.vet_contact, 15, ' ') ||
RPAD(vet_rec.vet_gender, 10, ' ')||
RPAD(vet_rec.transaction, 20, ' ') || 'RM '
||RPAD(TRIM(TO_CHAR(vet_rec.amount, '999G999G999D99')),
20, ' '));
v_totalTrans := v_totalTrans + vet_rec.transaction;
v_totalAmt := v_totalAmt + vet_rec.amount;

END LOOP;
DBMS_OUTPUT.PUT_LINE(LPAD('-', 95, '-'));
DBMS_OUTPUT.PUT_LINE(RPAD('*', 51, ' ') || 'Subtotal:'
||RPAD(v_totalTrans, 4, ' ') || RPAD(' ', 16, ' ') ||
'RM ' ||RPAD(TRIM(TO_CHAR(v_totalAmt,
'9G999G999G999D99')), 20, ' '));
DBMS_OUTPUT.PUT_LINE(LPAD('-', 95, '-'));
DBMS_OUTPUT.PUT_LINE(chr(10));

v_grandTrans := v_grandTrans + v_totalTrans;
v_grandTotal := v_grandTotal + v_totalAmt;

END LOOP;
DBMS_OUTPUT.PUT_LINE(LPAD('-', 95, '-'));
DBMS_OUTPUT.PUT_LINE(RPAD('*', 48, ' ') || 'Grand
Total:' ||RPAD(v_grandTrans, 5, ' ') || RPAD(' ', 15, '
') || 'RM ' ||RPAD(TRIM(TO_CHAR(v_grandTotal,
'9G999G999G999D99')), 25, ' ') || RPAD(' ', 3, ' '));
DBMS_OUTPUT.PUT_LINE(chr(10));

END;
/

exec PRC_VET_PERFORMANCE
```

Sample Output:



```
*                                                Report generated on : 31-08-2021 10:17:06by ADB

Branch              : PULAU PINANG
-----------------------------------------------------------------------------------------------
Staff ID  StaffName                 Staff Tel      Gender    Total Transactions  Total Amount
-----------------------------------------------------------------------------------------------
V0001     Nigel Ng                  0192933380     M         1454                RM 746,282.10
V0002     Michelle Lim              0164833390     F         1438                RM 744,810.00
V0005     Tan Yee Ru                0118765987     F         1419                RM 711,976.90
-----------------------------------------------------------------------------------------------
*                                                   Subtotal:4311                RM 2,203,069.00
-----------------------------------------------------------------------------------------------


Branch              : KUALA LUMPUR
-----------------------------------------------------------------------------------------------
Staff ID  StaffName                 Staff Tel      Gender    Total Transactions  Total Amount
-----------------------------------------------------------------------------------------------
V0003     Ng Yi Xuan                0124385103     F         1432                RM 735,224.20
V0004     Jason Ong                 0191238765     M         1382                RM 695,164.40
V0008     Cheah Su Ying             0169634532     F         1393                RM 713,879.20
-----------------------------------------------------------------------------------------------
*                                                   Subtotal:4207                RM 2,144,267.80
-----------------------------------------------------------------------------------------------
```



```
Branch              : KEDAH
-----------------------------------------------------------------------------------------------
Staff ID  StaffName                 Staff Tel      Gender    Total Transactions  Total Amount
-----------------------------------------------------------------------------------------------
V0006     Simon Tan                 0125784328     M         930                 RM 479,158.10
V0007     Edward Teoh               0178635698     M         950                 RM 481,190.00
V0009     Ooi Yen Chun              0110834761     M         952                 RM 489,793.80
-----------------------------------------------------------------------------------------------
*                                                   Subtotal:2832                RM 1,450,141.90
-----------------------------------------------------------------------------------------------


-----------------------------------------------------------------------------------------------
*                                                Grand Total:11350               RM 5,797,478.70


PL/SQL procedure successfully completed.
```

4.2.9 Report 2: Summary report of Supplier Profile

Purpose: The purpose of this report is to summarize the supplier information such as the company details, what product they supplied to us and the purchase transaction details with them. At the end, we can know the total transaction made and total stock we get supplied from the specific supplier.

SQL statement:

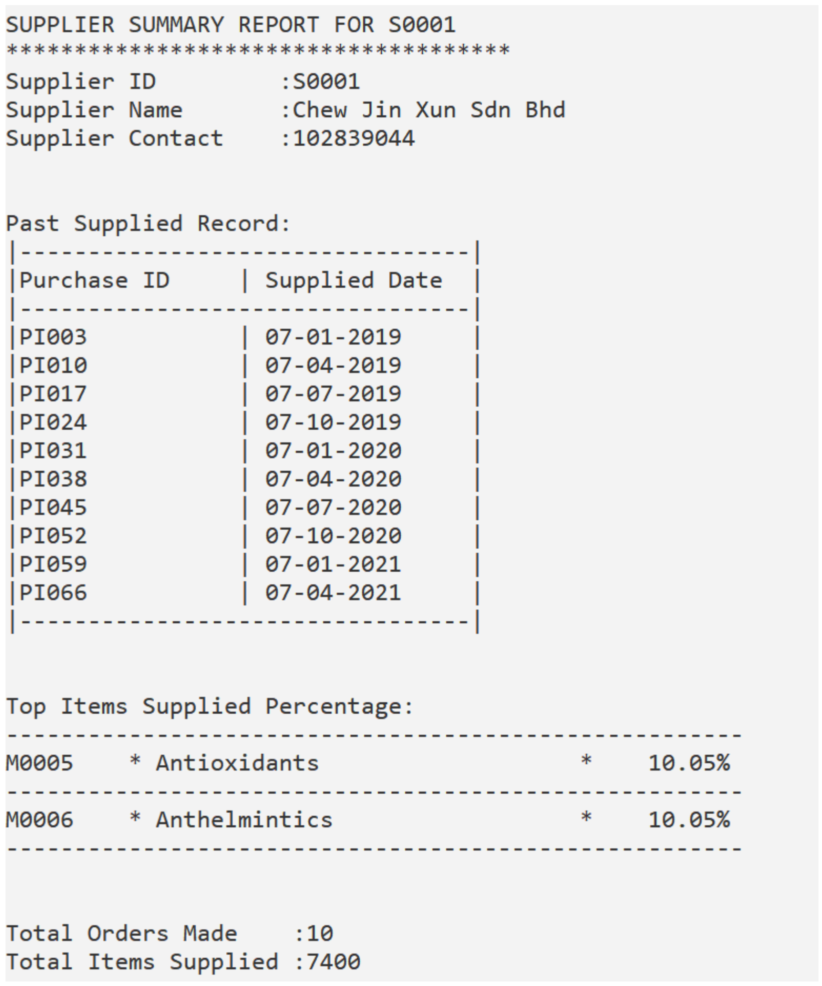
```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
create or replace procedure
prc_supplier_report(suppliercode in CHAR) Is
cursor purchaseTransaction_cursor Is
 select p.purchase_id, p.purchase_date, p.receive_date,
p.purchase_amount
 from purchaseTransaction p, supplier s
 where s.supplier_id = p.supplier_id
       and p.supplier_id = suppliercode;

cursor purchaseItem_cursor Is
 select m.medic_id, m.medic_name, sum(p.purchase_qty) as
quantity
 from medicalsupply m, purchaseItem p,
purchaseTransaction t
 where m.medic_id = p.medic_id
```

```
            and t.purchase_id = p.purchase_id
            and t.supplier_id = suppliercode
      group by m.medic_id, m.medic_name
      order by m.medic_id;

     EXCE_SUPPLIERCODE EXCEPTION;
     PRAGMA EXCEPTION_INIT(EXCE_SUPPLIERCODE, -20310);

     totalorder number(11,2);
     totalitem number(12,2):=0;
     totalqty number(12,2):=0;
     v_supplierID CHAR(6);
     v_supplierName VARCHAR2(30);
     v_suppliercontact NUMBER(15):=0;
     V_VALIDSUPPLIERID CHAR(6);
     V_prodrate NUMBER(5,2);

     BEGIN

         V_VALIDSUPPLIERID := SUPPLIERCODE;

         IF V_VALIDSUPPLIERID = ' ' THEN
           RAISE EXCE_SUPPLIERCODE;
         ELSE

         DBMS_OUTPUT.put_line(chr(10));
         DBMS_OUTPUT.PUT_LINE('SUPPLIER SUMMARY REPORT FOR
     '||UPPER(SUPPLIERCODE));

     DBMS_OUTPUT.PUT_LINE('**********************************
     ***');

         SELECT S.supplier_id, supplier_name,
     supplier_contact INTO
     v_supplierid,v_suppliername,v_suppliercontact
         FROM supplier S
         WHERE S.supplier_id = suppliercode;

         SELECT sum(purchase_qty) as TotalQty INTO TotalItem
         FROM purchaseItem;

         DBMS_OUTPUT.put_line(rpad('Supplier ID',20,'
     ')||':'||v_supplierid);
         DBMS_OUTPUT.put_line(rpad('Supplier Name',20,' ')||
     ':'||v_supplierName);
         DBMS_OUTPUT.put_line(rpad('Supplier Contact',20,'
     ')|| ':'||v_suppliercontact);
         DBMS_OUTPUT.put_line(chr(10));

         totalorder := 0;
         DBMS_OUTPUT.PUT_LINE('Past Supplied Record:');
```

```
    DBMS_OUTPUT.PUT_LINE('|'||rpad('-',33,'-')||'|');
    DBMS_OUTPUT.PUT_LINE('|'||rpad('Purchase ID',15,' ')
||' | '||rpad('Supplied Date',15,' ')||'|');
    DBMS_OUTPUT.PUT_LINE('|'||rpad('-',33,'-')||'|');

    for purch_rec in purchaseTransaction_cursor loop
     IF purch_rec.receive_date IS NULL THEN

DBMS_OUTPUT.PUT_LINE('|'||rpad(purch_rec.purchase_id,15,
' ') || ' | '||rpad('-',15,' ')||'|');
     ELSE

DBMS_OUTPUT.PUT_LINE('|'||rpad(purch_rec.purchase_id,15,
' ') || ' | '||rpad(purch_rec.receive_date,15,'
')||'|');
      totalorder := totalorder + 1;
     END IF;
    END LOOP;


    totalqty := 0;
    DBMS_OUTPUT.PUT_LINE('|'||rpad('-',33,'-')||'|');
    DBMS_OUTPUT.put_line(chr(10));
    DBMS_OUTPUT.PUT_LINE('Top Items Supplied
Percentage:');
    DBMS_OUTPUT.PUT_LINE(rpad('-',54,'-'));

    for purch_item in purchaseItem_cursor loop
     V_prodrate:=0;
     V_prodrate:= (purch_item.quantity/totalitem)*100;
     DBMS_OUTPUT.PUT_LINE(rpad(purch_item.medic_id,8,'
') ||' * '|| rpad(purch_item.medic_name,30,' ')||' * '
||lpad(V_prodrate,8,' ')||'%');
     DBMS_OUTPUT.PUT_LINE(rpad('-',54,'-'));
     totalqty := totalqty + purch_item.quantity;
    END LOOP;

  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE('Total Orders Made    :'
||totalorder);
  DBMS_OUTPUT.PUT_LINE('Total Items Supplied :'
||totalqty);

  END IF;
EXCEPTION
    WHEN EXCE_SUPPLIERCODE THEN
       DBMS_OUTPUT.PUT_LINE(-20310||'INVALID SUPPLIER
CODE.');
END;
/
```

```
exec PRC_SUPPLIER_REPORT('S0001');
```

Sample Output:



```
SUPPLIER SUMMARY REPORT FOR S0001
*************************************
Supplier ID        :S0001
Supplier Name      :Chew Jin Xun Sdn Bhd
Supplier Contact   :102839044


Past Supplied Record:
|---------------------------------|
|Purchase ID      | Supplied Date |
|---------------------------------|
|PI003            | 07-01-2019    |
|PI010            | 07-04-2019    |
|PI017            | 07-07-2019    |
|PI024            | 07-10-2019    |
|PI031            | 07-01-2020    |
|PI038            | 07-04-2020    |
|PI045            | 07-07-2020    |
|PI052            | 07-10-2020    |
|PI059            | 07-01-2021    |
|PI066            | 07-04-2021    |
|---------------------------------|


Top Items Supplied Percentage:
------------------------------------------------------
M0005    * Antioxidants                  *    10.05%
------------------------------------------------------
M0006    * Anthelmintics                 *    10.05%
------------------------------------------------------


Total Orders Made    :10
Total Items Supplied :7400
```

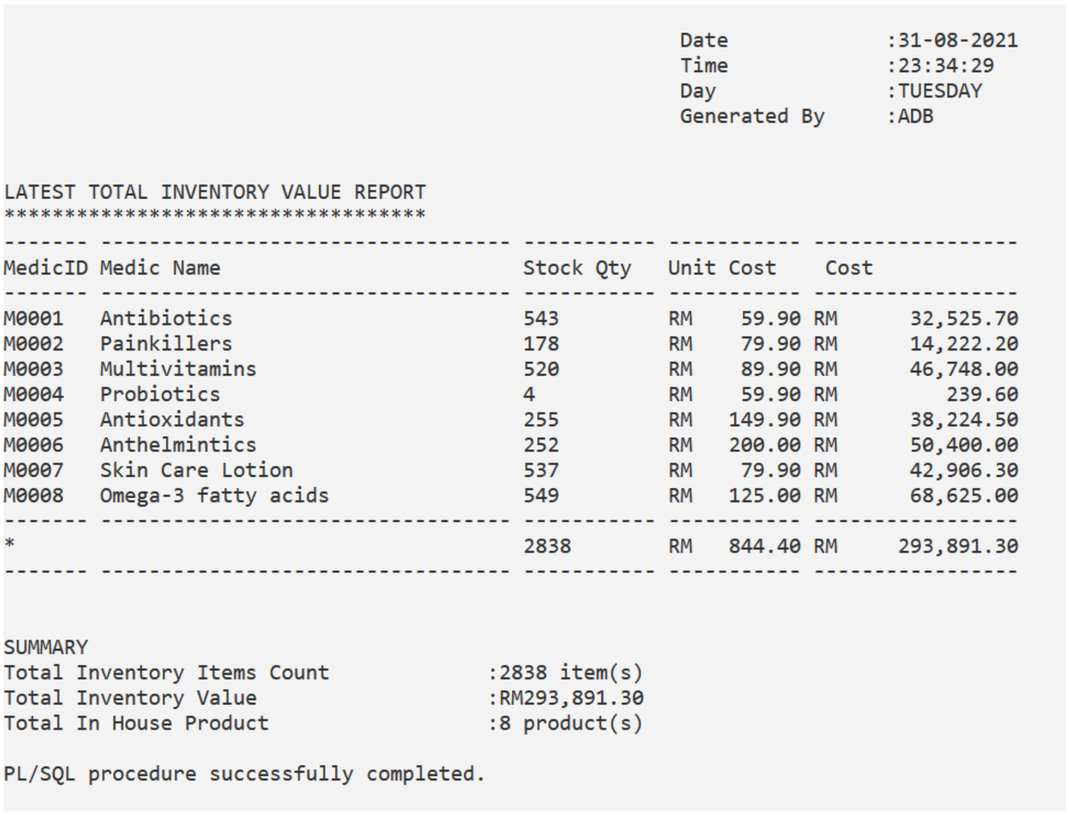
4.2.10 Report 3: On demand report of Inventory value
Purpose: The purpose of this report is to get the detail of our company inventory value such as how many stocks we left for every medicine and how much does it cost, which means the property company has in the warehouse. At the end, we can know the total amount of inventory and cost.

SQL statement:

```
set pagesize 1000
set linesize 200

create or replace procedure PRC_INVENTORY_REPORT is

 cursor medic_cursor Is
 select medic_id, medic_name, medic_qty, medic_price,
(medic_qty*medic_price) as amount
 from medicalsupply
 group by medic_id, medic_name, medic_qty, medic_price
 order by medic_id;
 medic_c medic_cursor% ROWTYPE;
 subtotal NUMBER;
 totalqty NUMBER;
 tunitcost NUMBER;
 tproduct NUMBER;
 grandqty NUMBER;
 grandtotal NUMBER;
BEGIN
DBMS_OUTPUT.put_line(chr(10));
DBMS_OUTPUT.PUT_LINE(rpad(chr(9),7,chr(9))||rpad('Date',
17,' ')||':'|| SYSDATE);
DBMS_OUTPUT.PUT_LINE(rpad(chr(9),7,chr(9))||rpad('Time',
17,' ')||':'||TO_CHAR(SYSDATE,'HH24:MI:SS'));
DBMS_OUTPUT.PUT_LINE(rpad(chr(9),7,chr(9))||rpad('Day',1
7,' ')||':'||TO_CHAR(SYSDATE,'DAY'));
DBMS_OUTPUT.PUT_LINE(rpad(chr(9),7,chr(9))||rpad('Genera
ted By',17,' ')||':' || USER);

DBMS_OUTPUT.put_line(chr(10));
DBMS_OUTPUT.PUT_LINE(' LATEST TOTAL INVENTORY VALUE
REPORT');
DBMS_OUTPUT.PUT_LINE('**********************************
*');

DBMS_OUTPUT.PUT_LINE(rpad('-',7,'-')||'
'||rpad('-',34,'-')||' '||rpad('-',11,'-')|| '
'||rpad('-',11,'-')||' '||rpad('-',17,'-'));
DBMS_OUTPUT.PUT_LINE(rpad('MedicID',8,' ')||rpad('Medic
Name',35,' ')|| rpad('Stock Qty',12,' ')|| rpad('Unit
Cost',13,' ')||rpad('Cost',15,' '));
DBMS_OUTPUT.PUT_LINE(rpad('-',7,'-')||'
'||rpad('-',34,'-')||' '||rpad('-',11,'-')|| '
'||rpad('-',11,'-')||' '||rpad('-',17,'-'));

 tproduct:=0;
 Subtotal:=0;
 totalqty:=0;
 tunitcost:=0;
 OPEN medic_cursor;
  LOOP
   FETCH medic_cursor INTO medic_c;
EXIT WHEN medic_cursor%NOTFOUND;
DBMS_OUTPUT.PUT_LINE(rpad(medic_c.medic_id,8,'
')||rpad(medic_c.medic_name,35,' ')||
rpad(medic_c.medic_qty,12,'
')||'RM'||lpad(TRIM(TO_CHAR(medic_c.medic_price,'999G999
D99')),9,' ')||' RM'||lpad(
TRIM(TO_CHAR(medic_c.Amount,'999G999D99')),15,' '));
subtotal:= subtotal + medic_c.Amount;
totalqty:=totalqty + medic_c.medic_qty;
tunitcost:=tunitcost+ medic_c.medic_price;
tproduct:= tproduct + 1;
```

```
END LOOP;
grandqty := totalqty;
grandtotal := subtotal;
    DBMS_OUTPUT.PUT_LINE(rpad('-',7,'-')||'
'||rpad('-',34,'-')||' '||rpad('-',11,'-')|| '
'||rpad('-',11,'-')||' '||rpad('-',17,'-'));
    DBMS_OUTPUT.PUT_LINE(rpad('*',43,'
')||rpad(totalqty,12,' ')||'RM'||
lpad(TRIM(TO_CHAR(tunitcost,'999G999D99')),9,' ')||'
RM'||lpad(TRIM(TO_CHAR(subtotal,'999G999G999D99')),15,'
'));
    DBMS_OUTPUT.PUT_LINE(rpad('-',7,'-')||'
'||rpad('-',34,'-')||' '||rpad('-',11,'-')|| '
'||rpad('-',11,'-')||' '||rpad('-',17,'-'));


DBMS_OUTPUT.put_line(chr(10));
DBMS_OUTPUT.put_line('SUMMARY');
DBMS_OUTPUT.PUT_LINE(rpad('Total Inventory Items Count
',40,' ')||':'||grandqty|| ' item(s)' );
DBMS_OUTPUT.PUT_LINE(rpad('Total Inventory Value ',40,'
')||':RM'||TRIM(TO_CHAR(grandtotal,'999G999G999D99')) );
DBMS_OUTPUT.PUT_LINE(rpad('Total In House Product ',40,'
')||':'||tproduct|| ' product(s)' );
END;
/

EXEC PRC_INVENTORY_REPORT
```

Sample Output:



```
                                                         Date             :31-08-2021
                                                         Time             :23:34:29
                                                         Day              :TUESDAY
                                                         Generated By     :ADB


LATEST TOTAL INVENTORY VALUE REPORT
***********************************
------- ---------------------------------- ----------- ----------- -----------------
MedicID Medic Name                         Stock Qty   Unit Cost    Cost
------- ---------------------------------- ----------- ----------- -----------------
M0001   Antibiotics                        543         RM    59.90 RM      32,525.70
M0002   Painkillers                        178         RM    79.90 RM      14,222.20
M0003   Multivitamins                      520         RM    89.90 RM      46,748.00
M0004   Probiotics                         4           RM    59.90 RM         239.60
M0005   Antioxidants                       255         RM   149.90 RM      38,224.50
M0006   Anthelmintics                      252         RM   200.00 RM      50,400.00
M0007   Skin Care Lotion                   537         RM    79.90 RM      42,906.30
M0008   Omega-3 fatty acids                549         RM   125.00 RM      68,625.00
------- ---------------------------------- ----------- ----------- -----------------
*                                          2838        RM   844.40 RM     293,891.30
------- ---------------------------------- ----------- ----------- -----------------


SUMMARY
Total Inventory Items Count             :2838 item(s)
Total Inventory Value                   :RM293,891.30
Total In House Product                  :8 product(s)

PL/SQL procedure successfully completed.
```

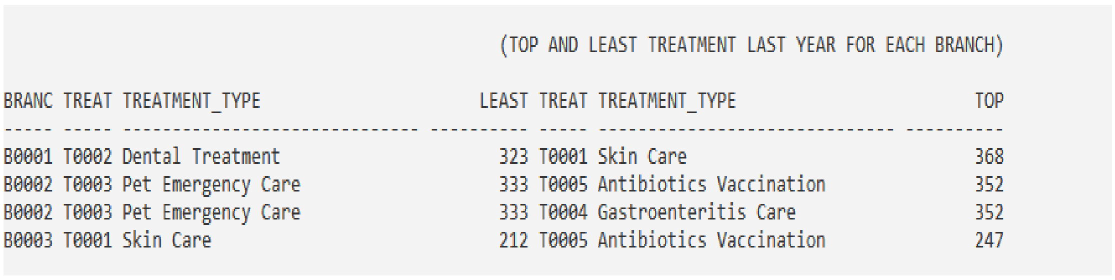
4.3 (Tan Wei Siong)

4.3.1 Query 1: Top and Least Treatment Last Year for Each Branch

Purpose: The purpose of this query is to let the manager to view the top and least treatment in the last year for each branch. With the query, the manager can determine which treatment is not famous in the branch. Thus, the manager can make the pricing strategy to promote their least treatment to their customer. The manager can also know the quantity difference between the most and the least treatment .

```
SET LINESIZE 150;
SET PAGESIZE 150;
COLUMN TREATMENT_TYPE FORMAT A30;
TTITLE CENTER ('TOP AND LEAST TREATMENT LAST YEAR FOR EACH BRANCH') SKIP 2

WITH outlets AS
(SELECT b.branch_id, t.treatment_id, t.treatment_type, COUNT(t.treatment_id) AS
Branch_total_qty
FROM   Treatment t, Appointment a, Transaction v, Branch b
WHERE  t.treatment_id = a.treatment_id AND
    a.appointment_id = v.appointment_id AND
    b.branch_id = v.branch_id        AND
    Extract(year from Appointment_datetime) = Extract(year from sysdate)-1
GROUP by b.branch_id, t.treatment_id, t.treatment_type
order by 1,4),
mn_trans AS
(SELECT branch_id, MIN(Branch_total_qty) AS Least
FROM outlets
group by branch_id
ORDER BY 1),
mx_trans AS
(SELECT branch_id,MAX(Branch_total_qty) AS Top
FROM outlets
group by branch_id
ORDER BY 1)
SELECT A.branch_id,C.treatment_id, C.treatment_type, A.Least,
D.treatment_id,D.treatment_type, B.Top
From mn_trans A JOIN mx_trans B ON (A.branch_id = B.branch_id)
   JOIN outlets C ON (A.branch_id = C.branch_id) AND c.Branch_total_qty = A.Least
   JOIN outlets D ON (A.branch_id = D.branch_id) AND D.Branch_total_qty = B.Top
ORDER BY 1;
```

Sample Output:



```
                                             (TOP AND LEAST TREATMENT LAST YEAR FOR EACH BRANCH)

BRANC TREAT TREATMENT_TYPE                      LEAST TREAT TREATMENT_TYPE                        TOP
----- ----- ------------------------------ ---------- ----- ------------------------------ ----------
B0001 T0002 Dental Treatment                      323 T0001 Skin Care                             368
B0002 T0003 Pet Emergency Care                    333 T0005 Antibiotics Vaccination               352
B0002 T0003 Pet Emergency Care                    333 T0004 Gastroenteritis Care                  352
B0003 T0001 Skin Care                             212 T0005 Antibiotics Vaccination               247
```

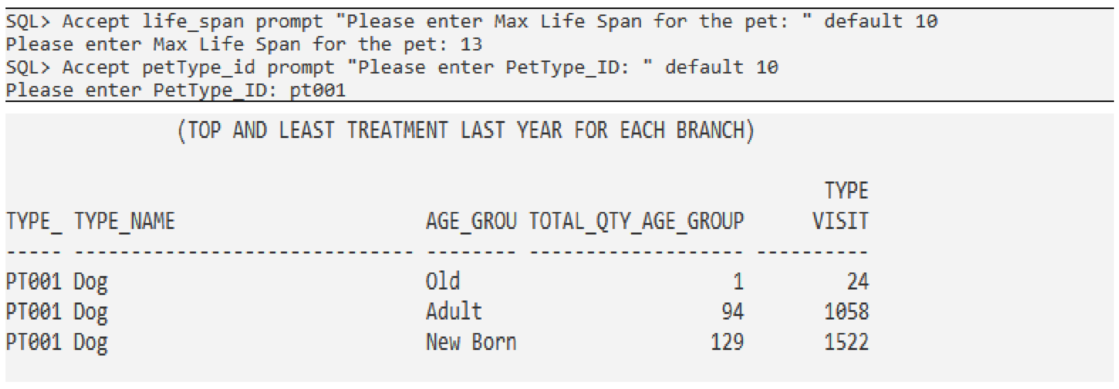
4.3.2 Query 2: Selected Pet Type Age Group and Respective Visit Quantity

Purpose: The purpose of this query is to find the selected pet type and the total visit quantity. With this query, the management can determine the number of selected pet type age groups and the total visit for the age group. The management can find the most number of the age group for the pet but having the low number of visits and propose the strategy to attract the age group. For instance, if the newborn pet dog has a large quantity but low visit, the management can make a promotion plan on the suitable treatment to attract the potential customer.

```
BREAK ON REPORT
TTITLE CENTER ('AGE GROUP PET AND TOTAL VISIT') SKIP 2
COMPUTE SUM OF Total_Visit ON REPORT;
COMPUTE SUM OF Total_Qty_Age_Group ON REPORT;
COLUMN type_id HEADING 'TYPE|ID';
COLUMN type_name HEADING 'TYPE|NAME';
COLUMN Total_Qty_Age_Group HEADING
'TOTAL|QTY|AGE|GROUP';
COLUMN Total_Visit HEADING 'TYPE|VISIT';
SET LINESIZE 80;

Accept life_span prompt "Please enter Max Life Span for
the pet: " default 10
```

```
Accept petType_id prompt "Please enter PetType_ID: "
default 10
WITH ageGroup as(
 SELECT CASE
   WHEN (EXTRACT (YEAR FROM sysdate) - EXTRACT(YEAR FROM
p.pet_dob) > ROUND(&life_span * 2/3)) THEN 'Old'
   WHEN (EXTRACT (YEAR FROM sysdate) - EXTRACT(YEAR FROM
p.pet_dob) > ROUND(&life_span * 1/3)) THEN 'Adult'
   ELSE 'New Born'
 END AS age_group, p.pet_id, p.type_id, t.type_name
 FROM  Pet p, PetType t
 WHERE p.type_id = t.type_id
 ORDER BY 1
),
 visitTime as (SELECT b.type_id, b.type_name,
b.age_group,
              COUNT(a.appointment_dateTime) AS
Total_Visit
 FROM  ageGroup b, Appointment a
 WHERE a.pet_id = b.pet_id
 GROUP BY type_id, type_name, age_group
),
 totalQtyAgeGroup as (SELECT type_id, type_name,
age_group, COUNT(age_group) As Total_QTY_Age_Group
 FROM  ageGroup
 GROUP BY type_id, type_name, age_group
)
 SELECT V.type_id, v.type_name, v.age_group,
t.Total_Qty_Age_Group, v.Total_Visit
 FROM   visitTime V
 LEFT JOIN totalQtyAgeGroup t ON V.type_id = t.type_id
AND t.age_group = v.age_group
 WHERE  V.type_id = UPPER('&petType_id')
 ORDER BY 1,2;
```

Sample Output:



```
SQL> Accept life_span prompt "Please enter Max Life Span for the pet: " default 10
Please enter Max Life Span for the pet: 13
SQL> Accept petType_id prompt "Please enter PetType_ID: " default 10
Please enter PetType_ID: pt001
```

```
               (TOP AND LEAST TREATMENT LAST YEAR FOR EACH BRANCH)

                                                                     TYPE
TYPE_ TYPE_NAME                      AGE_GROU TOTAL_QTY_AGE_GROUP     VISIT
----- ------------------------------ -------- ------------------- ----------
PT001 Dog                            Old                        1         24
PT001 Dog                            Adult                     94       1058
PT001 Dog                            New Born                 129       1522
```

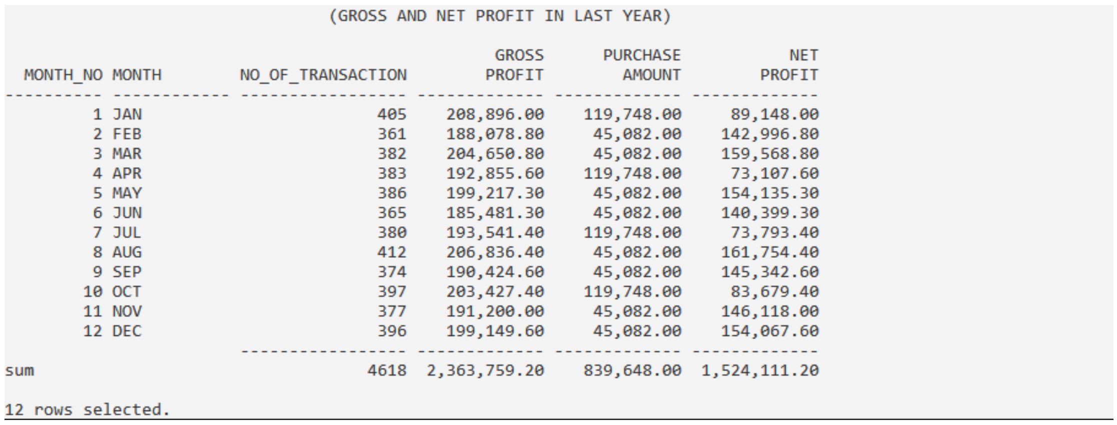
4.3.3 Query 3: Gross Profit and Net Profit Previous Year

Purpose: The purpose of this query is to view the gross profit and net profit in the previous year. The profit will be listed in each month so that the management can know the performance of the pet clinic. The gross profit is calculated based on the transaction for all branches and the purchase amount is the sum of the purchase medicine stock price. The net profit will be calculated by gross profit - purchase amount.

```
COLUMN GROSS_PROFIT FORMAT '9,999,999.99';
COLUMN PURCHASE_AMOUNT FORMAT '9,999,999.99';
COLUMN NET_PROFIT FORMAT '9,999,999.99';
COLUMN GROSS_PROFIT HEADING 'GROSS|PROFIT';
COLUMN PURCHASE_AMOUNT HEADING 'PURCHASE|AMOUNT';
COLUMN NET_PROFIT HEADING 'NET|PROFIT';
TTITLE CENTER ('GROSS AND NET PROFIT IN LAST YEAR') SKIP
2
SET LINESIZE 100;
SET PAGESIZE 200;
BREAK ON REPORT
COMPUTE SUM OF NET_PROFIT ON REPORT
COMPUTE SUM OF NO_Of_Transaction ON REPORT;
COMPUTE SUM OF GROSS_PROFIT ON REPORT;
COMPUTE SUM OF PURCHASE_AMOUNT ON REPORT;
COMPUTE SUM OF NET_PROFIT ON REPORT;

WITH last_year_transaction AS(
 SELECT transaction_id, transaction_dateTime
 FROM   Transaction
 WHERE  Extract(year from(transaction_dateTime)) =
extract(year from sysdate)-1),
last_year_purchase AS(
 SELECT purchase_id , purchase_Date, purchase_Amount
 FROM   PurchaseTransaction
 WHERE  Extract(year from(purchase_Date)) = Extract(year
from sysdate)-1),
purchase_detail AS(
 SELECT EXTRACT(Month from v.purchase_Date) AS Month_NO,
        TO_CHAR(v.purchase_Date,'MON') as Month,
        COUNT(v.purchase_id) AS NO_Of_Purchase,
        SUM(v.purchase_Amount) AS PURCHASE_AMOUNT
 FROM   last_year_purchase v
 GROUP BY EXTRACT (Month from v.purchase_Date),
          TO_CHAR(v.purchase_Date,'MON')
)
SELECT EXTRACT(Month from v.transaction_dateTime) AS
Month_NO,
       TO_CHAR(v.transaction_dateTime,'MON') as Month,
       COUNT(v.transaction_id) AS NO_Of_Transaction,
       SUM(t.total_amount) AS GROSS_PROFIT,
       l.purchase_Amount AS PURCHASE_AMOUNT,
       SUM(t.total_amount) - l.purchase_Amount AS
NET_PROFIT
```

```
FROM   last_year_transaction v, Transaction t
LEFT JOIN purchase_detail l ON l.Month_NO =
                              EXTRACT(Month from
transaction_dateTime)
WHERE  v.transaction_id = t.transaction_id
GROUP BY EXTRACT (Month from v.transaction_dateTime),
         TO_CHAR(v.transaction_dateTime,'MON'),
         l.purchase_Amount
ORDER BY 1;
```

Sample Output:



```
                              (GROSS AND NET PROFIT IN LAST YEAR)

                                                GROSS     PURCHASE           NET
  MONTH_NO MONTH        NO_OF_TRANSACTION      PROFIT       AMOUNT        PROFIT
---------- ------------ ----------------- ------------- ------------- -------------
         1 JAN                        405    208,896.00    119,748.00     89,148.00
         2 FEB                        361    188,078.80     45,082.00    142,996.80
         3 MAR                        382    204,650.80     45,082.00    159,568.80
         4 APR                        383    192,855.60    119,748.00     73,107.60
         5 MAY                        386    199,217.30     45,082.00    154,135.30
         6 JUN                        365    185,481.30     45,082.00    140,399.30
         7 JUL                        380    193,541.40    119,748.00     73,793.40
         8 AUG                        412    206,836.40     45,082.00    161,754.40
         9 SEP                        374    190,424.60     45,082.00    145,342.60
        10 OCT                        397    203,427.40    119,748.00     83,679.40
        11 NOV                        377    191,200.00     45,082.00    146,118.00
        12 DEC                        396    199,149.60     45,082.00    154,067.60
                        ----------------- ------------- ------------- -------------
sum                                  4618  2,363,759.20    839,648.00  1,524,111.20

12 rows selected.
```

4.3.4 Procedure 1: Pet Registration

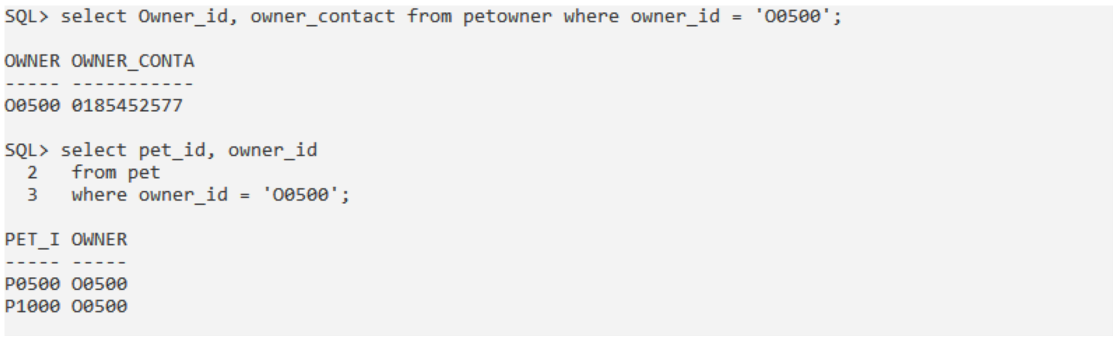
Purpose: The purpose of this procedure is to let the staff register the pet in an easier way. The staff is only required to input the owner's phone number and their pet information into this procedure and the data will be stored in the database. If the owner contact is not found, an error message will appear to alert the user to check the contact number input or ask them to register the pet owner first before registering the pet details.

Procedure code:

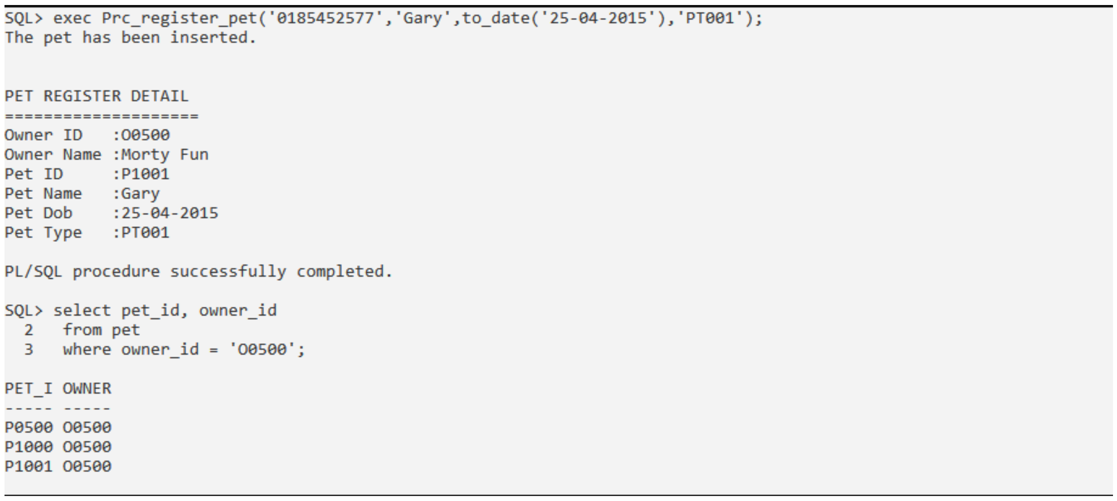
```
CREATE OR REPLACE Procedure
Prc_register_pet(in_owner_Contact IN VARCHAR2,
in_pet_Name IN VARCHAR2, in_pet_dob IN Date, in_pet_type
IN CHAR) AS
 No_owner_found EXCEPTION;
 PRAGMA exception_init(No_owner_found,-20201);
 v_petOwner_id  PetOwner.owner_id%TYPE;
 v_owner_name   PetOwner.owner_name%TYPE;
 v_pet_id       Pet.pet_id%TYPE;
 v_sequence     NUMBER;

BEGIN
 SELECT owner_id, owner_name INTO v_petOwner_id,
v_owner_name
```

```
 FROM   PetOwner
 WHERE  owner_contact = in_owner_Contact;

 IF SQL%FOUND THEN
  SELECT pet_seq.nextVal INTO v_sequence FROM dual;
  v_pet_id := TO_CHAR('P'||v_sequence);
  Insert into Pet values(v_pet_id, v_petOwner_id,
in_pet_Name, in_pet_dob, in_pet_type);
  dbms_output.put_line('The pet has been inserted.');
  dbms_output.put_line(chr(10));
  dbms_output.put_line('PET REGISTER DETAIL');
  dbms_output.put_line((LPAD('=',20,'=')));
  dbms_output.put_line('Owner ID   :'||v_petOwner_id);
  dbms_output.put_line('Owner Name :'||v_owner_name);
  dbms_output.put_line('Pet ID     :'||v_pet_id);
  dbms_output.put_line('Pet Name   :'||in_pet_Name);
  dbms_output.put_line('Pet Dob    :'||in_pet_dob);
  dbms_output.put_line('Pet Type   :'||in_pet_type);
 END IF;

EXCEPTION
   WHEN NO_DATA_FOUND THEN
   RAISE_application_error(-20201,'The owner is not
exist! Check Phone No or create new owner.');
END;
/
```

Sample Output:



```
SQL> select Owner_id, owner_contact from petowner where owner_id = 'O0500';

OWNER OWNER_CONTA
----- -----------
O0500 0185452577

SQL> select pet_id, owner_id
  2   from pet
  3   where owner_id = 'O0500';

PET_I OWNER
----- -----
P0500 O0500
P1000 O0500
```



```
SQL> exec Prc_register_pet('0185452577','Gary',to_date('25-04-2015'),'PT001');
The pet has been inserted.


PET REGISTER DETAIL
====================
Owner ID    :O0500
Owner Name :Morty Fun
Pet ID      :P1001
Pet Name    :Gary
Pet Dob     :25-04-2015
Pet Type    :PT001

PL/SQL procedure successfully completed.

SQL> select pet_id, owner_id
  2   from pet
  3   where owner_id = 'O0500';

PET_I OWNER
----- -----
P0500 O0500
P1000 O0500
P1001 O0500
```

Exception

```
SQL> exec Prc_register_pet('01199887766','Gary',to_date('25-04-2015'),'PT001');
BEGIN Prc_register_pet('01199887766','Gary',to_date('25-04-2015'),'PT001'); END;

*
ERROR at line 1:
ORA-20201: The owner is not exist! Check Phone No or create new owner.
ORA-06512: at "ADB.PRC_REGISTER_PET", line 32
ORA-06512: at line 1
```

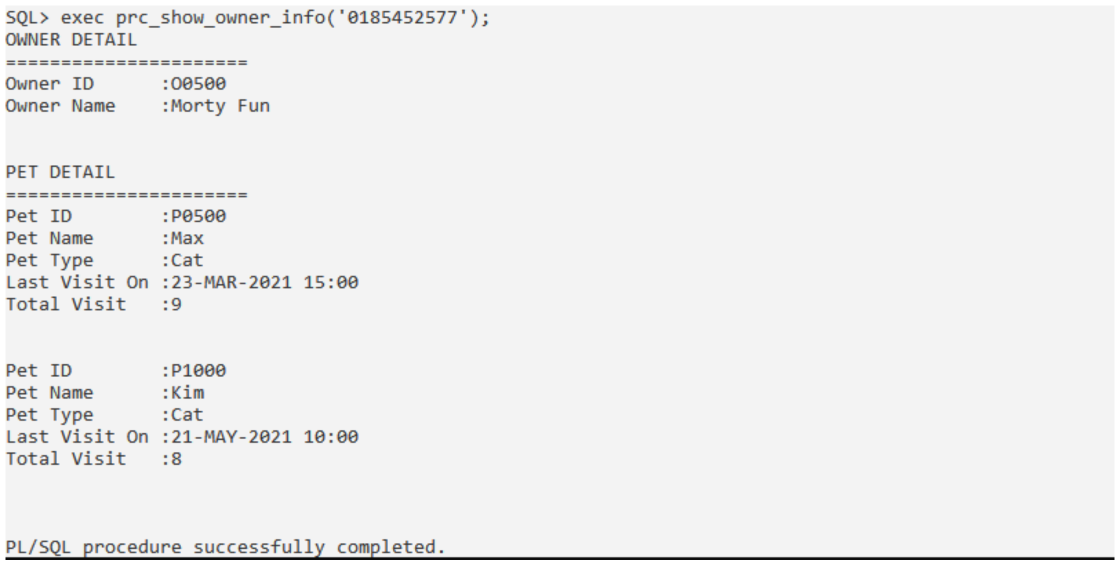
4.3.5 Procedure 2: Display Owner Information

Purpose: The purpose of this procedure is to list the owner's pet information. With this procedure, the staff can know all the pet details of the owner. For instance, last visit date for the pet and total visit quantity.

Procedure code:
```
CREATE OR REPLACE Procedure
Prc_show_owner_info(in_owner_Contact IN VARCHAR2) AS
 CURSOR owner_list IS
  SELECT o.owner_id, o.owner_name, p.pet_id, p.pet_name,
         t.type_name AS pet_Type,
MAX(appointment_datetime) AS last_Vist,
         COUNT(appointment_datetime) AS Total_Visit
  FROM   PetOwner o, Pet p, Appointment a, PetType t
  WHERE  o.owner_contact = 0185452577 AND
         a.pet_id   = p.pet_id              AND
         t.type_id  = p.type_id             AND
         o.owner_id = p.owner_id
  Group by o.owner_id, o.owner_name, p.pet_id,
p.pet_name, t.type_name
  Order by 3;

 owner_r owner_list%ROWTYPE;
  v_count  NUMBER;

BEGIN
 v_count := 0;
 OPEN owner_list;
  LOOP
    FETCH owner_list INTO owner_r;
    EXIT WHEN owner_list%NOTFOUND;
     IF (v_count = 0) THEN
       dbms_output.put_line('     OWNER DETAIL');
       dbms_output.put_line((LPAD('=',22,'=')));
       dbms_output.put_line('Owner ID
:'||owner_r.owner_id);
       dbms_output.put_line('Owner Name
:'||owner_r.owner_name);
       dbms_output.put_line(chr(10));
       dbms_output.put_line('      PET DETAIL');
       dbms_output.put_line((LPAD('=',22,'=')));
     END IF;
       dbms_output.put_line('Pet ID
:'||owner_r.pet_id);
       dbms_output.put_line('Pet Name
:'||owner_r.pet_name);
       dbms_output.put_line('Pet Type
:'||owner_r.pet_Type);
       dbms_output.put_line('Last Visit On
:'||owner_r.last_Vist);
```

```
       dbms_output.put_line('Total Visit
:'||owner_r.Total_Visit);
       dbms_output.put_line(chr(10));
     v_count := v_count + 1;
  END LOOP;
 CLOSE owner_list;
END;
/
```

Sample Output:



```
SQL> exec prc_show_owner_info('0185452577');
OWNER DETAIL
======================
Owner ID      :O0500
Owner Name    :Morty Fun


PET DETAIL
======================
Pet ID        :P0500
Pet Name      :Max
Pet Type      :Cat
Last Visit On :23-MAR-2021 15:00
Total Visit   :9


Pet ID        :P1000
Pet Name      :Kim
Pet Type      :Cat
Last Visit On :21-MAY-2021 10:00
Total Visit   :8


PL/SQL procedure successfully completed.
```

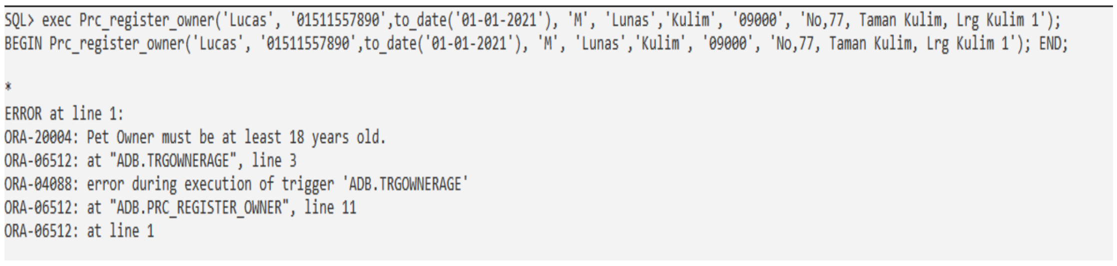
4.3.6 Trigger 1: Check Owner Age

Purpose: The purpose of this trigger is to check the owner's age. If the owner is less than 18, this trigger will occur and generate the error message and disallow the data insert into the database.

Trigger code:

```
CREATE OR REPLACE TRIGGER trgOwnerAge
  BEFORE INSERT OR UPDATE ON PetOwner
  FOR EACH ROW
BEGIN
  IF((ROUND((SYSDATE-:new.owner_dob )/365)) < 18) THEN
    RAISE_APPLICATION_ERROR(-20004, 'Pet Owner must be
at least 18 years old.' );
  END IF;
END;
/
```

Sample Output:



```
SQL> exec Prc_register_owner('Lucas', '01511557890',to_date('01-01-2021'), 'M', 'Lunas','Kulim', '09000', 'No,77, Taman Kulim, Lrg Kulim 1');
BEGIN Prc_register_owner('Lucas', '01511557890',to_date('01-01-2021'), 'M', 'Lunas','Kulim', '09000', 'No,77, Taman Kulim, Lrg Kulim 1'); END;

*
ERROR at line 1:
ORA-20004: Pet Owner must be at least 18 years old.
ORA-06512: at "ADB.TRGOWNERAGE", line 3
ORA-04088: error during execution of trigger 'ADB.TRGOWNERAGE'
ORA-06512: at "ADB.PRC_REGISTER_OWNER", line 11
ORA-06512: at line 1
```

4.3.7 Trigger 2: Check Appointment Date Time

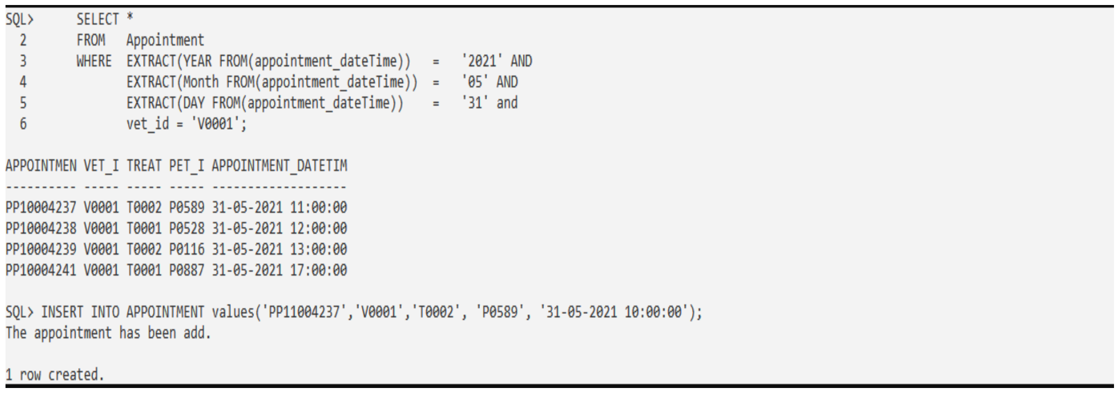
Purpose: The purpose of this trigger is to check the appointment date time. If the selected date and time for the selected vet has been appointed, this trigger will not allow the appointment to be made and prompt the suggested time to the user.

Trigger code:

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY
HH24:MI:SS';
CREATE OR REPLACE TRIGGER TRG_CHK_APPOINTMENT_DATE
BEFORE INSERT ON Appointment
FOR EACH ROW
DECLARE
   Date_Time_Booked EXCEPTION;
   PRAGMA exception_init(Date_Time_Booked, -20200 );
   v_tempDate             DATE;
   v_startDate            DATE;
   v_BOOKED_APPOINTMENT   DATE;
   v_appointment_dateTime DATE;
   v_count        NUMBER;
   v_check        NUMBER;
   v_cursorLength NUMBER;

    CURSOR c1 IS
     SELECT appointment_dateTime
     FROM   Appointment
     WHERE  EXTRACT(YEAR FROM(appointment_dateTime))    =
EXTRACT(YEAR FROM(:new.appointment_dateTime)) AND
            EXTRACT(Month FROM(appointment_dateTime))   =
EXTRACT(Month FROM(:new.appointment_dateTime)) AND
            EXTRACT(DAY FROM(appointment_dateTime))     =
EXTRACT(DAY FROM(:new.appointment_dateTime)) AND
            vet_id = :new.vet_id;

    CURSOR c2 IS
     SELECT appointment_dateTime
     FROM   Appointment
     WHERE  EXTRACT(YEAR FROM(appointment_dateTime))    =
EXTRACT(YEAR FROM(:new.appointment_dateTime)) AND
            EXTRACT(Month FROM(appointment_dateTime))   =
EXTRACT(Month FROM(:new.appointment_dateTime)) AND
            EXTRACT(DAY FROM(appointment_dateTime))     =
EXTRACT(DAY FROM(:new.appointment_dateTime)) AND
            vet_id = :new.vet_id;

BEGIN
 v_startDate     := TRUNC(:new.appointment_dateTime);
 v_count         := 0;
 v_cursorLength  := 0;
 SELECT a.appointment_dateTime into
V_appointment_dateTime
 FROM   Appointment a
 WHERE  a.appointment_dateTime  =
:new.appointment_dateTime AND
        a.vet_id = :new.vet_id;
 IF SQL%FOUND THEN
  dbms_output.put_line('The time has been book.');
     OPEN c1;
        LOOP
         FETCH c1 INTO v_tempDate;
          EXIT WHEN c1%NOTFOUND;
          v_cursorLength := v_cursorLength+1;
        END LOOP;
     IF v_cursorLength > 7 THEN
        dbms_output.put_line('The vet is fulled on this
date. Please choose other vet or change date.');
        RAISE_application_error(-20200,'The Date Time
has been booked');
     END IF;
  v_startDate := v_startDate + 10/24;
  dbms_output.put_line('Suggested Date Time');
  dbms_output.put_line('===================');
  WHILE v_count <= 7
    LOOP
       v_check := 0;
       OPEN c2;
        LOOP
        FETCH c2 INTO v_booked_appointment;
         EXIT WHEN c2%NOTFOUND;
          IF (v_startDate != v_booked_appointment AND
v_check <= v_cursorLength) THEN
            v_check := v_check + 1;
          END IF;
        END LOOP;
       CLOSE c2;
        IF v_check = v_cursorLength THEN
          dbms_output.put_line(v_startDate);
        END IF;
     v_count := v_count + 1;
     v_startDate := v_startDate + 1/24;
    END LOOP;
   dbms_output.put_line('===================');
   dbms_output.put_line('Available time slot on '||
v_startDate ||' for the selected vet is shown above.' );
 RAISE_application_error(-20200,'The Date Time has been
booked');
 END IF;
 EXCEPTION
 WHEN NO_DATA_FOUND THEN
  dbms_output.put_line('The appointment has been add.');
```

```
END;
/
```

Sample Output:



```
SQL>      SELECT *
          FROM   Appointment
          WHERE  EXTRACT(YEAR FROM(appointment_dateTime))   =   '2021' AND
                 EXTRACT(Month FROM(appointment_dateTime))  =   '05' AND
                 EXTRACT(DAY FROM(appointment_dateTime))    =   '31' and
                 vet_id = 'V0001';

APPOINTMEN VET_I TREAT PET_I APPOINTMENT_DATETIM
---------- ----- ----- ----- -------------------
PP10004237 V0001 T0002 P0589 31-05-2021 11:00:00
PP10004238 V0001 T0001 P0528 31-05-2021 12:00:00
PP10004239 V0001 T0002 P0116 31-05-2021 13:00:00
PP10004241 V0001 T0001 P0887 31-05-2021 17:00:00

SQL> INSERT INTO APPOINTMENT values('PP11004237','V0001','T0002', 'P0589', '31-05-2021 10:00:00');
The appointment has been add.

1 row created.
```

4.3.8 Report 1: Summary Report of Treatment Revenue in Selected Year

Purpose: The purpose of this summary report is to show the treatment revenue for the selected year. This report can help the management to understand the total treatment quantity and revenue for each branch in the selected year. This report can help the management to know which treatment is most famous or profitable in each branch. This report can help the management to gain the insight of the treatment for each branch.

Report code:

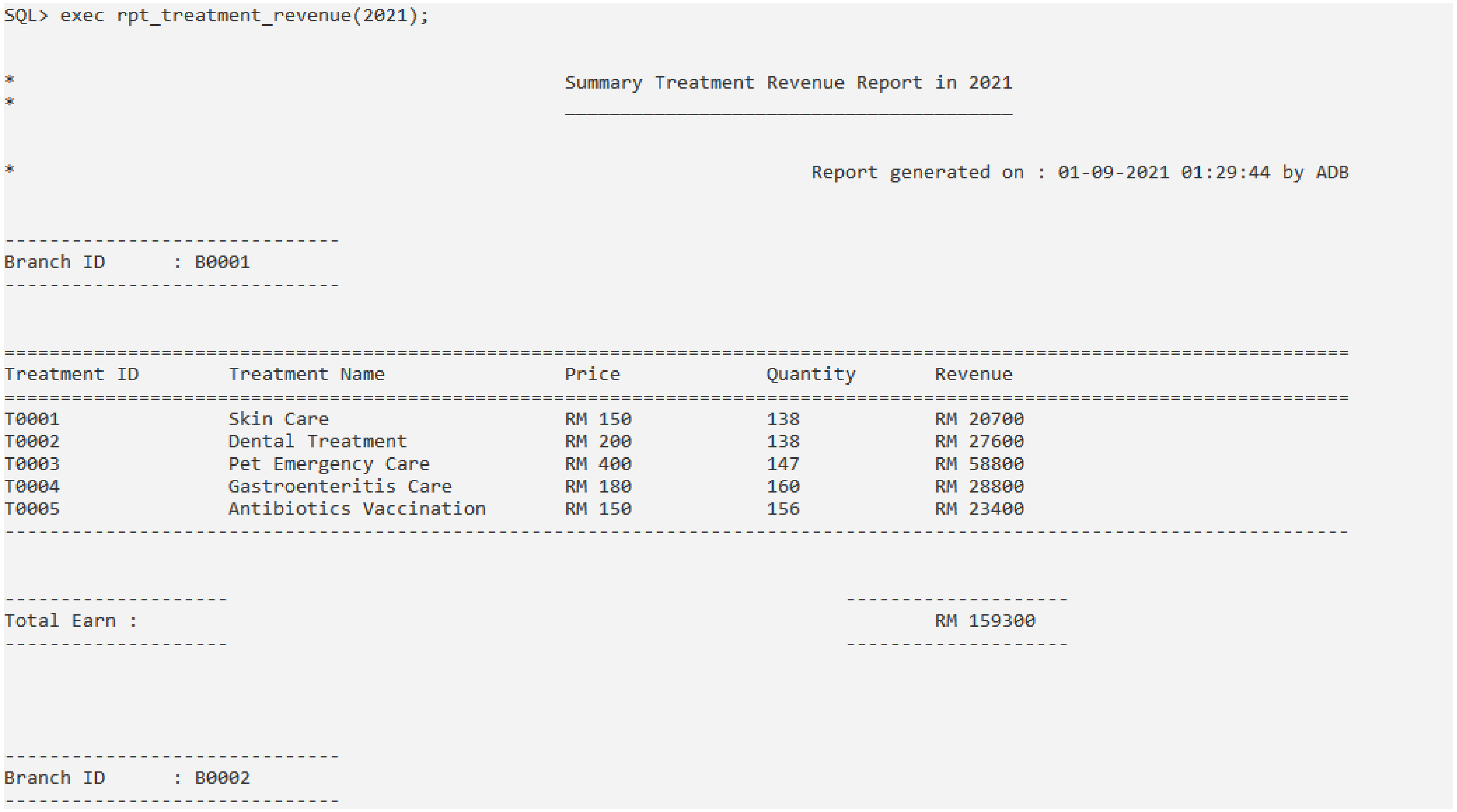
```
set linesize 175;
set pagesize 200;
 e_invalid_year EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_year, -20166);

 CURSOR pet_detail_list IS
CREATE OR REPLACE PROCEDURE
rpt_treatment_revenue(IN_YEAR IN NUMBER) AS
CURSOR treatment_list IS
 SELECT b.branch_id, t.treatment_id, t.treatment_type as
treatment_name,
        t.treatment_price as Price,

                                COUNT(t.treatment_price) as qty,

          SUM(t.treatment_price) as Total_Treatment_Earn
 FROM   Treatment t, Appointment a, Branch b,
Transaction v
 WHERE  b.branch_id = v.branch_id  AND
        v.appointment_id = a.appointment_id  AND

                            a.treatment_id = t.treatment_id  AND
```

```
    EXTRACT(YEAR FROM(v.transaction_dateTime)) = IN_YEAR
GROUP BY b.branch_id, t.treatment_id, t.treatment_type,
         t.treatment_price
ORDER BY 1;

treatment_list_r treatment_list%ROWTYPE;
v_count  NUMBER;
v_revenue  NUMBER;
v_total_revenue  NUMBER;
v_branch_id VARCHAR2(5);

BEGIN
IF IN_YEAR > Extract(Year From(sysdate)+1) or IN_YEAR <
2019 THEN
 RAISE_application_error(-20166,'Invalid year');
END IF;
 DBMS_OUTPUT.PUT_LINE(chr(10));
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 50, ' ') || 'Summary
Treatment Revenue Report in '|| IN_YEAR);
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 50, ' ') || RPAD('_',
40,'_'));
 DBMS_OUTPUT.PUT_LINE(chr(10));
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 72, ' ') || 'Report
generated on : ' ||
                      TO_CHAR(CURRENT_DATE, 'DD-MM-YYYY
HH:MI:SS ') ||




                                          'by ' || USER);
 DBMS_OUTPUT.PUT_LINE(chr(10));
 v_count := 0;
 v_revenue :=0;
 v_total_revenue :=0;
 v_branch_id := ' ';
 OPEN treatment_list;
  LOOP
   FETCH treatment_list INTO treatment_list_r;
    EXIT WHEN treatment_list%NOTFOUND;
                       IF v_count > 0 AND v_branch_id !=
treatment_list_r.branch_id THEN
              DBMS_OUTPUT.PUT_LINE(LPAD('-', 120, '-'));
     DBMS_OUTPUT.PUT_LINE(chr(10));
     DBMS_OUTPUT.PUT_LINE(RPAD('--------------------',
75, ' ')  ||'--------------------');
     DBMS_OUTPUT.PUT_LINE(RPAD('Total Earn :', 83, ' ')
||'RM ' || v_revenue);
     DBMS_OUTPUT.PUT_LINE(RPAD('--------------------',
75, ' ')  ||'--------------------');
```

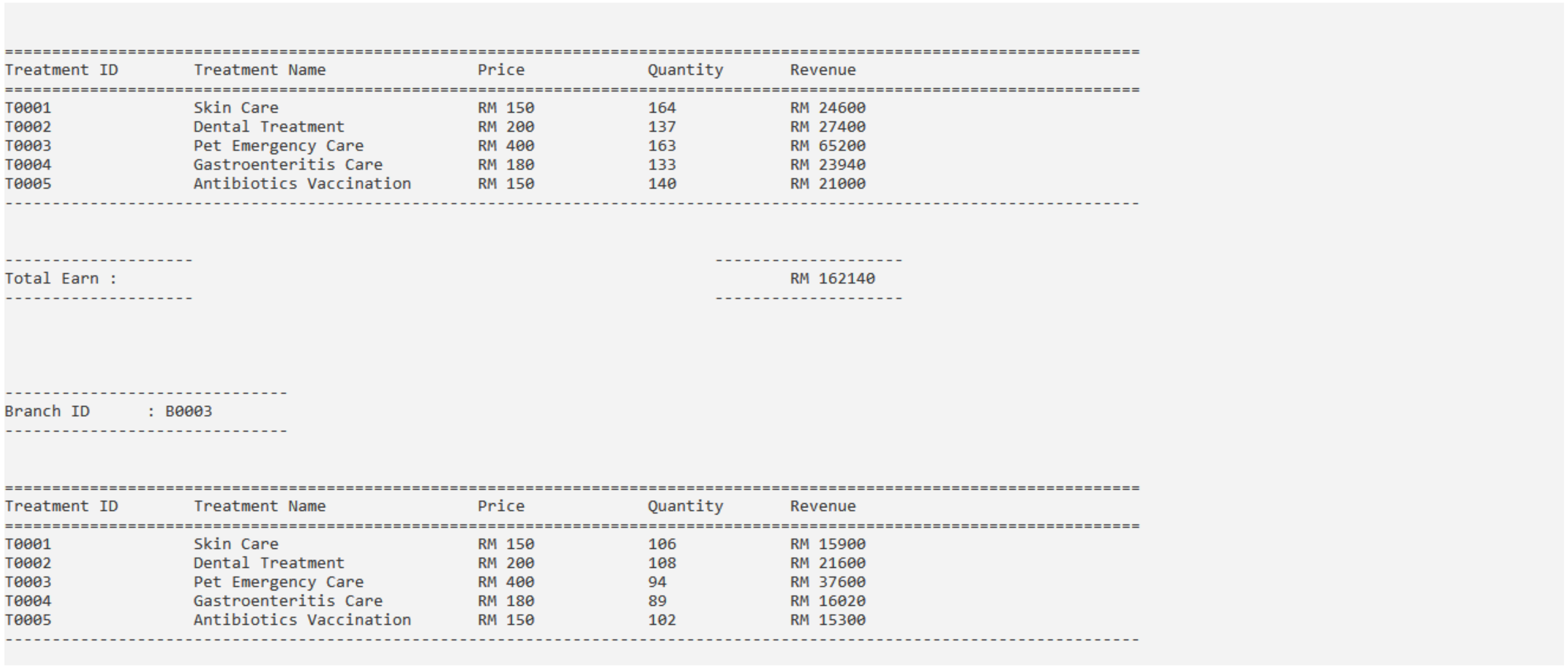
```
          DBMS_OUTPUT.PUT_LINE(chr(10));
                                 DBMS_OUTPUT.PUT_LINE(chr(10));
              v_total_revenue := v_total_revenue + v_revenue;
                                                 v_revenue :=0;
                                                        END IF;
                          IF v_count = 0 OR v_branch_id !=
treatment_list_r.branch_id THEN
                      DBMS_OUTPUT.PUT_LINE(LPAD('-', 30, '-'));
         DBMS_OUTPUT.PUT_LINE(RPAD('Branch ID', 15, ' ') ||
': ' ||

RPAD(UPPER(treatment_list_r.branch_id), 10, ' '));
                      DBMS_OUTPUT.PUT_LINE(LPAD('-', 30, '-'));
                                 DBMS_OUTPUT.PUT_LINE(chr(10));
                     DBMS_OUTPUT.PUT_LINE(LPAD('=', 120, '='));
        DBMS_OUTPUT.PUT_LINE(RPAD('Treatment ID', 20, ' ') ||
                                RPAD('Treatment Name', 30, ' ')
||




                                       RPAD('Price',18, ' ') ||




                                    RPAD('Quantity',15, ' ') ||




                                        RPAD('Revenue',20, ' ')




                                                             );
                     DBMS_OUTPUT.PUT_LINE(LPAD('=', 120, '='));
                                                        END IF;
DBMS_OUTPUT.PUT_LINE(RPAD(treatment_list_r.treatment_id,
20, ' ')||
             RPAD(treatment_list_r.treatment_name, 30, ' ')||

              'RM ' ||RPAD(treatment_list_r.Price, 15, ' ')||

                        RPAD(treatment_list_r.qty, 15, ' ')||

 'RM ' ||RPAD(treatment_list_r.Total_Treatment_Earn, 20, ' ')
                                                         );
                                     v_count := v_count + 1;
                                    v_revenue := v_revenue +
    treatment_list_r.Total_Treatment_Earn;
                v_branch_id := treatment_list_r.branch_id;
      END LOOP;
                DBMS_OUTPUT.PUT_LINE(LPAD('-', 120, '-'));
         DBMS_OUTPUT.PUT_LINE(chr(10));
         DBMS_OUTPUT.PUT_LINE(RPAD('--------------------',
    77, ' ')  ||'--------------------');
         DBMS_OUTPUT.PUT_LINE(RPAD('Total Earn :', 83, ' ')
    ||'RM ' || v_revenue);
         DBMS_OUTPUT.PUT_LINE(RPAD('--------------------',
    77, ' ')  ||'--------------------');
         DBMS_OUTPUT.PUT_LINE(chr(10));
             v_total_revenue := v_total_revenue + v_revenue;
      DBMS_OUTPUT.PUT_LINE(RPAD('========================',
    77, ' ') ||'====================');
         DBMS_OUTPUT.PUT_LINE(RPAD('Total Treatment Revenue:
    ', 83, ' ') || 'RM ' || v_total_revenue);

    DBMS_OUTPUT.PUT_LINE(RPAD('========================',
    77, ' ') ||'====================');
                                DBMS_OUTPUT.PUT_LINE(chr(10));
            DBMS_OUTPUT.PUT_LINE(RPAD('*', 92, ' ')||'End of
    report');
    END;
```

```
/
```

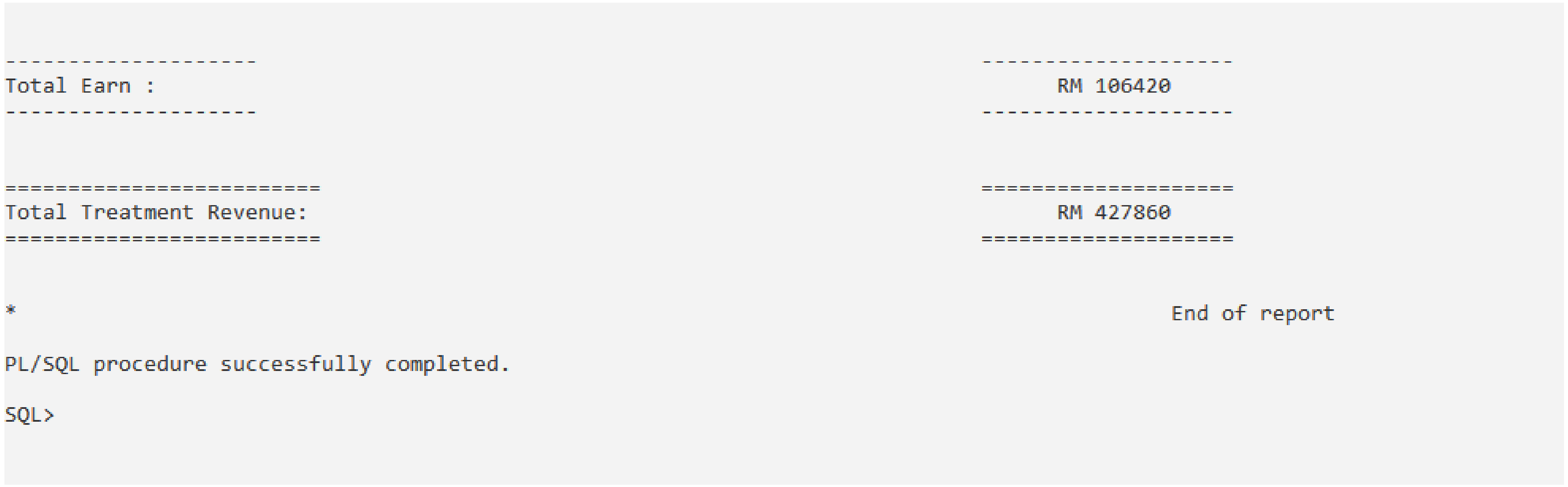
Sample Output:



```
SQL> exec rpt_treatment_revenue(2021);


*                                                 Summary Treatment Revenue Report in 2021
*                                                 ________________________________________


*                                                                      Report generated on : 01-09-2021 01:29:44 by ADB


------------------------------
Branch ID      : B0001
------------------------------


========================================================================================================================
Treatment ID        Treatment Name                Price             Quantity       Revenue
========================================================================================================================
T0001               Skin Care                     RM 150            138            RM 20700
T0002               Dental Treatment              RM 200            138            RM 27600
T0003               Pet Emergency Care            RM 400            147            RM 58800
T0004               Gastroenteritis Care          RM 180            160            RM 28800
T0005               Antibiotics Vaccination       RM 150            156            RM 23400
------------------------------------------------------------------------------------------------------------------------


--------------------                                                       --------------------
Total Earn :                                                                       RM 159300
--------------------                                                       --------------------




------------------------------
Branch ID      : B0002
------------------------------
```



```
========================================================================================================================
Treatment ID        Treatment Name               Price            Quantity       Revenue
========================================================================================================================
T0001               Skin Care                    RM 150           164            RM 24600
T0002               Dental Treatment             RM 200           137            RM 27400
T0003               Pet Emergency Care           RM 400           163            RM 65200
T0004               Gastroenteritis Care         RM 180           133            RM 23940
T0005               Antibiotics Vaccination      RM 150           140            RM 21000
------------------------------------------------------------------------------------------------------------------------


--------------------                                                          --------------------
Total Earn :                                                                          RM 162140
--------------------                                                          --------------------




------------------------------
Branch ID      : B0003
------------------------------


========================================================================================================================
Treatment ID        Treatment Name               Price            Quantity       Revenue
========================================================================================================================
T0001               Skin Care                    RM 150           106            RM 15900
T0002               Dental Treatment             RM 200           108            RM 21600
T0003               Pet Emergency Care           RM 400           94             RM 37600
T0004               Gastroenteritis Care         RM 180           89             RM 16020
T0005               Antibiotics Vaccination      RM 150           102            RM 15300
------------------------------------------------------------------------------------------------------------------------
```



```
--------------------                                                              --------------------
Total Earn :                                                                            RM 106420
--------------------                                                              --------------------


=========================                                                         ====================
Total Treatment Revenue:                                                                RM 427860
=========================                                                         ====================

*                                                                                               End of report

PL/SQL procedure successfully completed.

SQL>
```

4.3.9 Report 2: Detail Report of Medic Performance

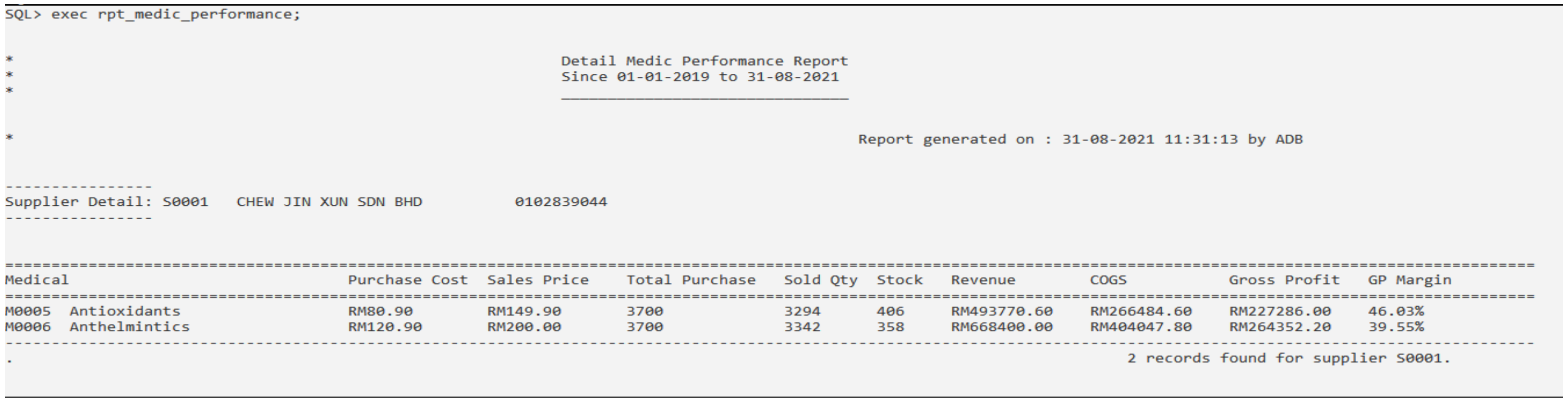
Purpose: The purpose of this detailed report is to show all medic performance from the start of the business until today. It will show the total purchase, total sales, stock quantity, gross profit and gross profit margin for each medicine. This report can help the management to know the performance of their medicine.
The calculation is show below:

Stock Quantity = Purchase quantity - sold quantity
Revenue = Sales price * sold quantity
Cost of good sold(COSG) = Purchase price * sold quantity
Gross profit = Revenue - COSG
Gross profit margin = Gross profit / Revenue * 100

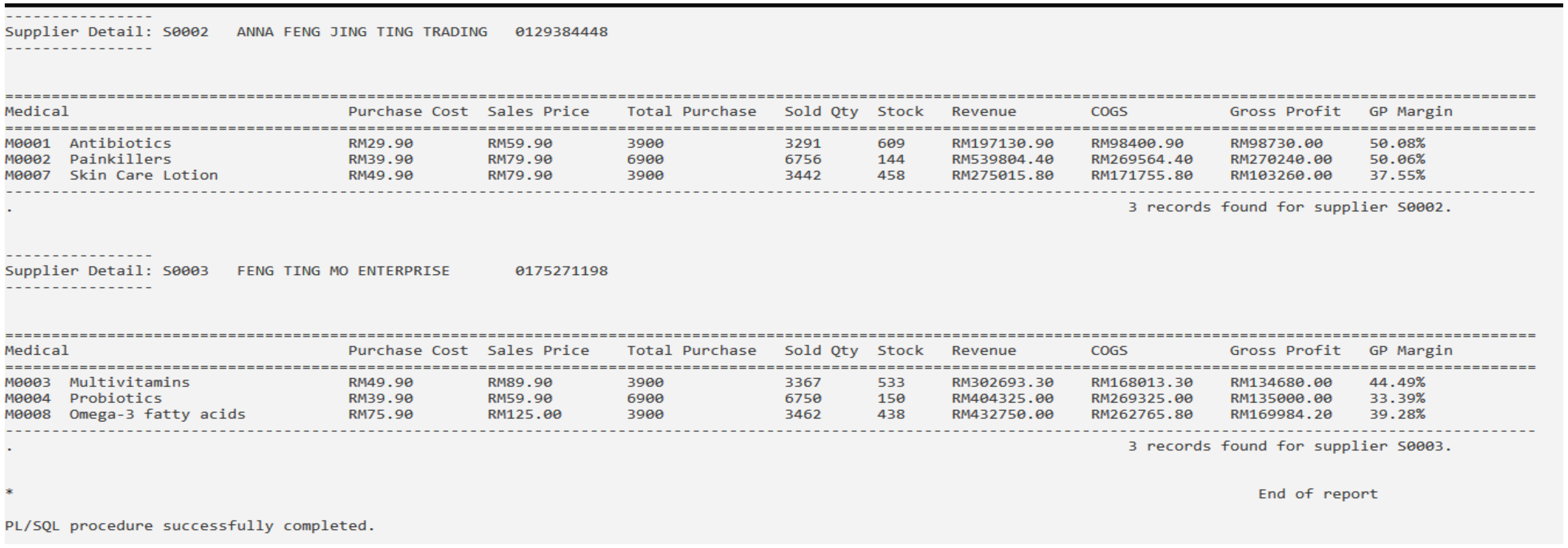
Report code:
```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY';
set linesize 170;
set pagesize 200;

CREATE OR REPLACE PROCEDURE rpt_medic_performance AS
CURSOR medic_list IS
 SELECT s.supplier_id, s.supplier_name,
s.supplier_contact, m.medic_id,
        m.medic_name, i.purchase_price,
SUM(i.purchase_qty) as Purchase_qty,

     m.medic_price as Sales_price, m.medic_qty as stock,

          (SUM(i.purchase_qty)- m.medic_qty) as Sold_qty
 FROM   Supplier s, PurchaseTransaction p, PurchaseItem
i,
        MedicalSupply m
 WHERE  s.supplier_id = p.supplier_id  AND
        p.purchase_id = i.purchase_id AND

                                 i.medic_id = m.medic_id
 GROUP BY s.supplier_id, s.supplier_name,
s.supplier_contact,
          m.medic_id, m.medic_name, i.purchase_price,

                              m.medic_price, m.medic_qty
 ORDER BY 1;
 medic_list_r medic_list%ROWTYPE;
 v_supplier_id VARCHAR(5);
 v_count NUMBER;
 v_count2 NUMBER;
 grossProfit NUMBER(11,2);
 grossProfitMargin NUMBER(11,2);
 revenue NUMBER(15,2);
 COGS NUMBER(15,2);
 totalGrossProfit NUMBER(15,2);

BEGIN
 v_count := 0;
 v_count2 := 0;
 grossProfit := 0;
 grossProfitMargin := 0;
 revenue := 0;
 COGS := 0;
 v_supplier_id := ' ';
 totalGrossProfit := 0;

 DBMS_OUTPUT.PUT_LINE(chr(10));
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 60, ' ') || 'Detail
Medic Performance Report');
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 60, ' ') || 'Since
01-01-2019' ||' to '|| sysdate);
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 60, ' ') || RPAD('_',
31,'_'));
 DBMS_OUTPUT.PUT_LINE(chr(10));
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 92, ' ') || 'Report
generated on : ' ||
                      TO_CHAR(CURRENT_DATE, 'DD-MM-YYYY
HH:MI:SS ') ||



                                             'by ' || USER);
 DBMS_OUTPUT.PUT_LINE(chr(10));

 OPEN medic_list;
  LOOP
   FETCH medic_list INTO medic_list_r;
    EXIT WHEN medic_list%NOTFOUND;
                     IF v_count > 0 AND v_supplier_id !=
medic_list_r.supplier_id THEN
              DBMS_OUTPUT.PUT_LINE(LPAD('-', 165, '-'));
       DBMS_OUTPUT.PUT_LINE('.'||LPAD(v_count2||'
records found for supplier ' || v_supplier_id|| '.',
155, ' '));
                          DBMS_OUTPUT.PUT_LINE(chr(10));
                                  totalgrossprofit := 0;
                                          v_count2 := 0;
                                                 END IF;
                      IF v_count = 0 OR v_supplier_id !=
medic_list_r.supplier_id THEN
               DBMS_OUTPUT.PUT_LINE(LPAD('-', 16, '-'));
    DBMS_OUTPUT.PUT_LINE(RPAD('Supplier Detail', 15, '
') || ': ' ||
RPAD(UPPER(medic_list_r.supplier_id), 8, ' ') ||

        RPAD(UPPER(medic_list_r.supplier_name), 30, ' ') ||

    RPAD(UPPER(medic_list_r.supplier_contact), 15, ' '));
                DBMS_OUTPUT.PUT_LINE(LPAD('-', 16, '-'));
                           DBMS_OUTPUT.PUT_LINE(chr(10));
               DBMS_OUTPUT.PUT_LINE(LPAD('=', 165, '='));
          DBMS_OUTPUT.PUT_LINE(RPAD('Medical', 37, ' ') ||

                            RPAD('Purchase Cost',15, ' ') ||

                              RPAD('Sales Price',15, ' ') ||

                           RPAD('Total Purchase',17, ' ') ||

                                 RPAD('Sold Qty',10, ' ') ||

                                     RPAD('Stock',8, ' ') ||
                                        RPAD('Revenue',15, ' ') ||



                                           RPAD('COGS',15, ' ') ||



                                   RPAD('Gross Profit',15, ' ') ||



                                      RPAD('GP Margin %',10, ' ')




                                                                );
                    DBMS_OUTPUT.PUT_LINE(LPAD('=', 165, '='));
                                                           END IF;

                         revenue := medic_list_r.Sales_price*
medic_list_r.Sold_qty;
                        COGS := medic_list_r.purchase_price *
medic_list_r.Sold_qty;
                              grossProfit := revenue - COGS;
       grossProfitMargin := grossProfit / revenue * 100;

DBMS_OUTPUT.PUT_LINE(RPAD(medic_list_r.medic_id, 7, ' ')
||
                               RPAD(medic_list_r.medic_name,
30, ' ') ||




   RPAD('RM'||(TRIM(TO_CHAR(medic_list_r.purchase_price,
'9999D99'))), 15, ' ') ||
          RPAD('RM'||(TRIM(TO_CHAR(medic_list_r.Sales_price,
'9999D99'))), 15, ' ') ||

                RPAD(medic_list_r.purchase_qty,17, ' ') ||

                    RPAD(medic_list_r.Sold_qty,10, ' ') ||

                        RPAD(medic_list_r.stock,8, ' ') ||

        RPAD('RM'||(TRIM(TO_CHAR(revenue, '999999999D99'))),
15, ' ') ||

  RPAD('RM'||(TRIM(TO_CHAR(COGS, '999999999D99'))), 15, '
') ||

                   RPAD('RM'||(TRIM(TO_CHAR(grossprofit,
'999999999D99'))), 15, ' ') ||

  RPAD(TRIM(TO_CHAR(grossProfitMargin, '999D99')) || '%',
10, ' ')
```

```
                                                      );
                                  v_count := v_count + 1;
                                v_count2 := v_count2 + 1;
                                            revenue := 0;
                                               COGS := 0;
                 v_supplier_id := medic_list_r.supplier_id;

  END LOOP;
  DBMS_OUTPUT.PUT_LINE(LPAD('-', 165, '-'));
  DBMS_OUTPUT.PUT_LINE('.'||LPAD(v_count2||' records
found for supplier ' || v_supplier_id|| '.', 155, ' '));
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(RPAD('*', 135, ' ')||'End of
report');

END;
/
```

Sample Output:



```
SQL> exec rpt_medic_performance;

*                                                           Detail Medic Performance Report
*                                                           Since 01-01-2019 to 31-08-2021
*                                                           _______________________________


*                                                                                            Report generated on : 31-08-2021 11:31:13 by ADB


---------------
Supplier Detail: S0001   CHEW JIN XUN SDN BHD         0102839044
---------------


==================================================================================================================================================================
Medical                             Purchase Cost  Sales Price    Total Purchase    Sold Qty  Stock   Revenue        COGS           Gross Profit   GP Margin
==================================================================================================================================================================
M0005  Antioxidants                 RM80.90        RM149.90       3700              3294      406     RM493770.60    RM266484.60    RM227286.00    46.03%
M0006  Anthelmintics                RM120.90       RM200.00       3700              3342      358     RM668400.00    RM404047.80    RM264352.20    39.55%
------------------------------------------------------------------------------------------------------------------------------------------------------------------
.                                                                                                                   2 records found for supplier S0001.
```



```
---------------
Supplier Detail: S0002   ANNA FENG JING TING TRADING   0129384448
---------------


==================================================================================================================================================================
Medical                             Purchase Cost  Sales Price    Total Purchase    Sold Qty  Stock   Revenue        COGS           Gross Profit   GP Margin
==================================================================================================================================================================
M0001  Antibiotics                  RM29.90        RM59.90        3900              3291      609     RM197130.90    RM98400.90     RM98730.00     50.08%
M0002  Painkillers                  RM39.90        RM79.90        6900              6756      144     RM539804.40    RM269564.40    RM270240.00    50.06%
M0007  Skin Care Lotion             RM49.90        RM79.90        3900              3442      458     RM275015.80    RM171755.80    RM103260.00    37.55%
------------------------------------------------------------------------------------------------------------------------------------------------------------------
.                                                                                                                   3 records found for supplier S0002.


---------------
Supplier Detail: S0003   FENG TING MO ENTERPRISE       0175271198
---------------


==================================================================================================================================================================
Medical                             Purchase Cost  Sales Price    Total Purchase    Sold Qty  Stock   Revenue        COGS           Gross Profit   GP Margin
==================================================================================================================================================================
M0003  Multivitamins                RM49.90        RM89.90        3900              3367      533     RM302693.30    RM168013.30    RM134680.00    44.49%
M0004  Probiotics                   RM39.90        RM59.90        6900              6750      150     RM404325.00    RM269325.00    RM135000.00    33.39%
M0008  Omega-3 fatty acids          RM75.90        RM125.00       3900              3462      438     RM432750.00    RM262765.80    RM169984.20    39.28%
------------------------------------------------------------------------------------------------------------------------------------------------------------------
.                                                                                                                   3 records found for supplier S0003.


*                                                                                                                                End of report

PL/SQL procedure successfully completed.
```

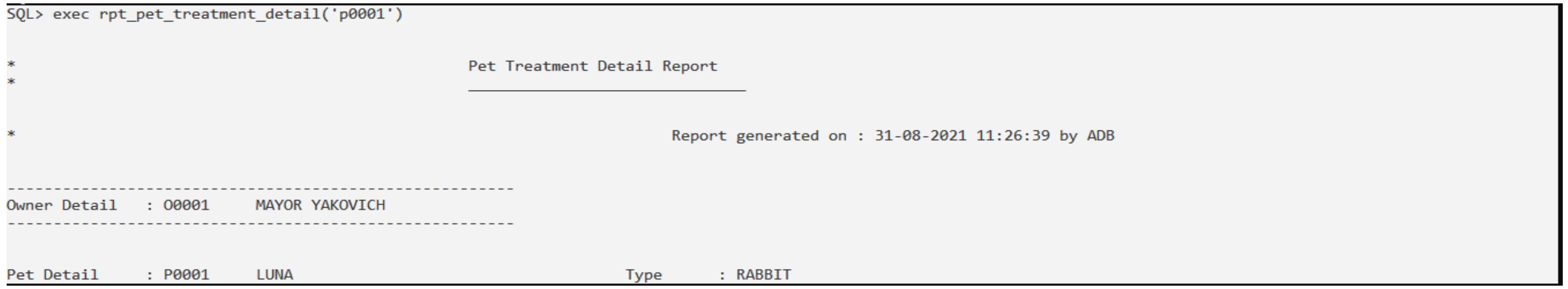
4.3.10 Report 3: On-demand Report of the Pet Treatment Detail

Purpose: The purpose of this on-demand report is to let the veterinarian view the pet treatment history. This report will show all treatment, treatment date and handle veterinarian for the selected pet. It can help the veterinarian to know the condition of the pet and provide the most suitable treatment or medicine for it.

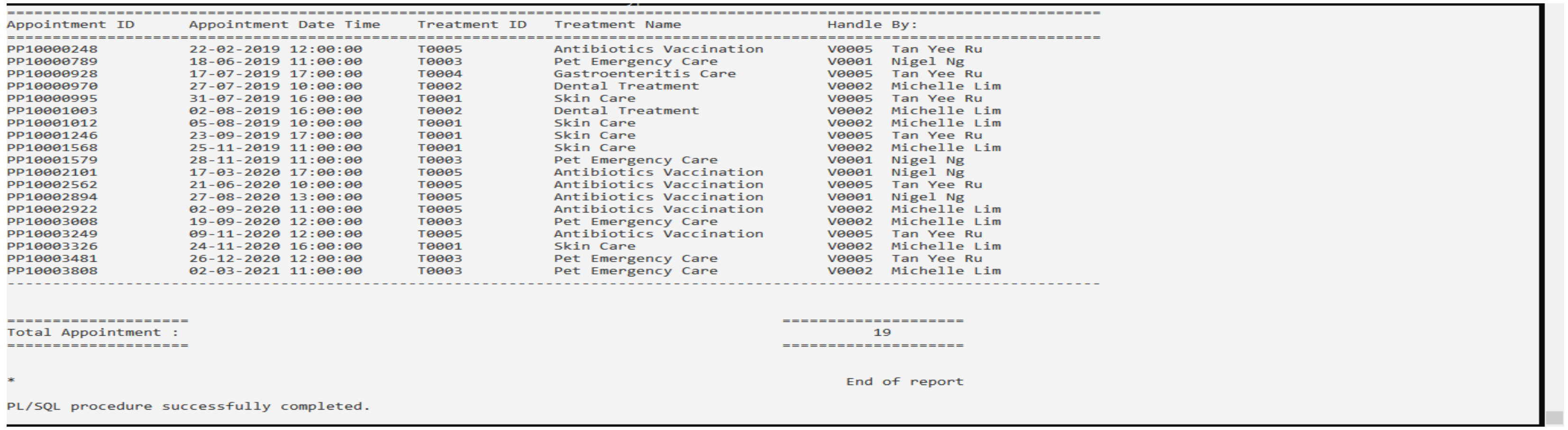
Report code:

```
ALTER SESSION SET NLS_DATE_FORMAT = 'DD-MM-YYYY
HH24:MI:SS';
set linesize 125;
CREATE OR REPLACE PROCEDURE
rpt_pet_treatment_detail(IN_PETID IN CHAR) AS

 NO_PET_FOUND EXCEPTION;
 PRAGMA EXCEPTION_INIT(NO_PET_FOUND, -20202);
 CURSOR pet_detail_list IS
  SELECT o.owner_id, o.owner_name, p.pet_id,
         p.pet_name,t.type_name AS pet_Type,

                                           a.appointment_id,

                 a.appointment_datetime, m.treatment_id,

           m.treatment_type AS treatment_name, v.vet_id,
v.vet_name
  FROM    PetOwner o, Pet p, PetType t,
         Veterinarian v,Appointment a,

                                                 Treatment m
  WHERE  o.owner_id = p.owner_id AND
         p.type_id = t.type_id   AND

                             p.pet_id = a.pet_id     AND

                      p.pet_id = UPPER(IN_PETID)     AND
         a.vet_id = v.vet_id     AND
         m.treatment_id = a.treatment_id
  ORDER BY 7;

  pet_detail_r pet_detail_list%ROWTYPE;
  v_count  NUMBER;

BEGIN
 DBMS_OUTPUT.PUT_LINE(chr(10));
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 50, ' ') || 'Pet
Treatment Detail Report');
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 50, ' ') || RPAD('_',
30,'_'));
 DBMS_OUTPUT.PUT_LINE(chr(10));
 DBMS_OUTPUT.PUT_LINE(RPAD('*', 72, ' ') || 'Report
generated on : ' ||
                     TO_CHAR(CURRENT_DATE, 'DD-MM-YYYY
HH:MI:SS ') ||




                                            'by ' || USER);
 DBMS_OUTPUT.PUT_LINE(chr(10));

 v_count := 0;
 OPEN pet_detail_list;
  LOOP
   FETCH pet_detail_list INTO pet_detail_r;
    EXIT WHEN pet_detail_list%NOTFOUND;
   IF (v_count = 0) THEN
    DBMS_OUTPUT.PUT_LINE(LPAD('-', 55, '-'));
    DBMS_OUTPUT.PUT_LINE(RPAD('Owner Detail', 15, ' ')
|| ': ' ||
                      RPAD(UPPER(pet_detail_r.owner_id),
10, ' ') ||





         RPAD(UPPER(pet_detail_r.owner_name), 40, ' '));
               DBMS_OUTPUT.PUT_LINE(LPAD('-', 55, '-'));
                          DBMS_OUTPUT.PUT_LINE(chr(10));
 DBMS_OUTPUT.PUT_LINE(RPAD('Pet Detail', 15, ' ' ) || ':
' ||
                       RPAD(UPPER(pet_detail_r.pet_id),
10, ' ')||





           RPAD(UPPER(pet_detail_r.pet_name), 40, ' ')||





                        RPAD('Type', 10, ' ' ) || ': ' ||
            RPAD(UPPER(pet_detail_r.pet_Type), 20, ' '));
               DBMS_OUTPUT.PUT_LINE(LPAD('=', 120, '='));
     DBMS_OUTPUT.PUT_LINE(RPAD('Appointment ID', 20, ' ') ||
                              RPAD('Appointment Date Time',
25, ' ') ||

                           RPAD('Treatment ID',15, ' ') ||

                          RPAD('Treatment Name',30, ' ') ||

                               RPAD('Handle By:',40, ' ')

                                                        );
               DBMS_OUTPUT.PUT_LINE(LPAD('=', 120, '='));
      END IF;
     DBMS_OUTPUT.PUT_LINE(RPAD(pet_detail_r.appointment_id,
20, ' ')||

RPAD(pet_detail_r.appointment_datetime, 25, ' ')||

              RPAD(pet_detail_r.treatment_id, 15, ' ')||

             RPAD(pet_detail_r.treatment_name, 30, ' ')||
                    RPAD(pet_detail_r.vet_id, 7, ' ')||



                   RPAD(pet_detail_r.vet_name, 30, ' ')




                                                      );
                                 v_count := v_count + 1;
  END LOOP;
  IF v_count = 0 THEN
  RAISE_application_error(-20202,'No pet found');
  END IF;

  DBMS_OUTPUT.PUT_LINE(LPAD('-', 120, '-'));
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(RPAD('====================', 85,
' ')  ||'====================');
  DBMS_OUTPUT.PUT_LINE(RPAD('Total Appointment :', 95, '
') || v_count);
  DBMS_OUTPUT.PUT_LINE(RPAD('====================', 85,
' ') ||'====================');
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(RPAD('*', 92, ' ')||'End of
report');


END;
/
```

Sample Output:

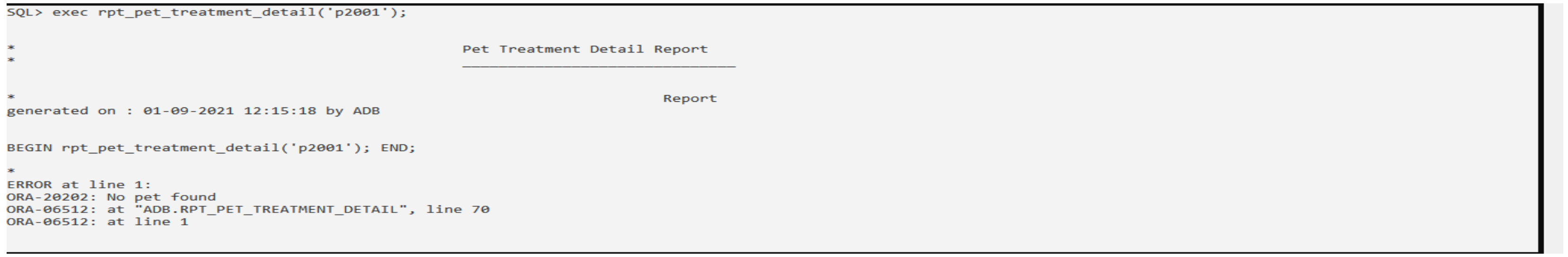


```
SQL> exec rpt_pet_treatment_detail('p0001')

*                                              Pet Treatment Detail Report
*                                              ____________________________

*                                                                    Report generated on : 31-08-2021 11:26:39 by ADB

-------------------------------------------------------
Owner Detail    : O0001     MAYOR YAKOVICH
-------------------------------------------------------

Pet Detail      : P0001     LUNA                                Type       : RABBIT
```



```
========================================================================================================================
Appointment ID      Appointment Date Time    Treatment ID   Treatment Name                 Handle By:
========================================================================================================================
PP10000248          22-02-2019 12:00:00      T0005          Antibiotics Vaccination        V0005  Tan Yee Ru
PP10000789          18-06-2019 11:00:00      T0003          Pet Emergency Care             V0001  Nigel Ng
PP10000928          17-07-2019 17:00:00      T0004          Gastroenteritis Care           V0005  Tan Yee Ru
PP10000970          27-07-2019 10:00:00      T0002          Dental Treatment               V0002  Michelle Lim
PP10000995          31-07-2019 16:00:00      T0001          Skin Care                      V0005  Tan Yee Ru
PP10001003          02-08-2019 16:00:00      T0002          Dental Treatment               V0002  Michelle Lim
PP10001012          05-08-2019 10:00:00      T0001          Skin Care                      V0002  Michelle Lim
PP10001246          23-09-2019 17:00:00      T0001          Skin Care                      V0005  Tan Yee Ru
PP10001568          25-11-2019 11:00:00      T0001          Skin Care                      V0002  Michelle Lim
PP10001579          28-11-2019 11:00:00      T0003          Pet Emergency Care             V0001  Nigel Ng
PP10002101          17-03-2020 17:00:00      T0005          Antibiotics Vaccination        V0001  Nigel Ng
PP10002562          21-06-2020 10:00:00      T0005          Antibiotics Vaccination        V0005  Tan Yee Ru
PP10002894          27-08-2020 13:00:00      T0005          Antibiotics Vaccination        V0001  Nigel Ng
PP10002922          02-09-2020 11:00:00      T0005          Antibiotics Vaccination        V0002  Michelle Lim
PP10003008          19-09-2020 12:00:00      T0003          Pet Emergency Care             V0002  Michelle Lim
PP10003249          09-11-2020 12:00:00      T0005          Antibiotics Vaccination        V0005  Tan Yee Ru
PP10003326          24-11-2020 16:00:00      T0001          Skin Care                      V0002  Michelle Lim
PP10003481          26-12-2020 12:00:00      T0003          Pet Emergency Care             V0005  Tan Yee Ru
PP10003808          02-03-2021 11:00:00      T0003          Pet Emergency Care             V0002  Michelle Lim
------------------------------------------------------------------------------------------------------------------------


====================                                                                              ====================
Total Appointment :                                                                                        19
====================                                                                              ====================

*                                                                                                      End of report

PL/SQL procedure successfully completed.
```

Exception:



```
SQL> exec rpt_pet_treatment_detail('p2001');

*                                                   Pet Treatment Detail Report
*                                                   ____________________________

*                                                                                 Report
generated on : 01-09-2021 12:15:18 by ADB


BEGIN rpt_pet_treatment_detail('p2001'); END;

*
ERROR at line 1:
ORA-20202: No pet found
ORA-06512: at "ADB.RPT_PET_TREATMENT_DETAIL", line 70
ORA-06512: at line 1
```

4.4 (Nigel Lee Jian Hsee)

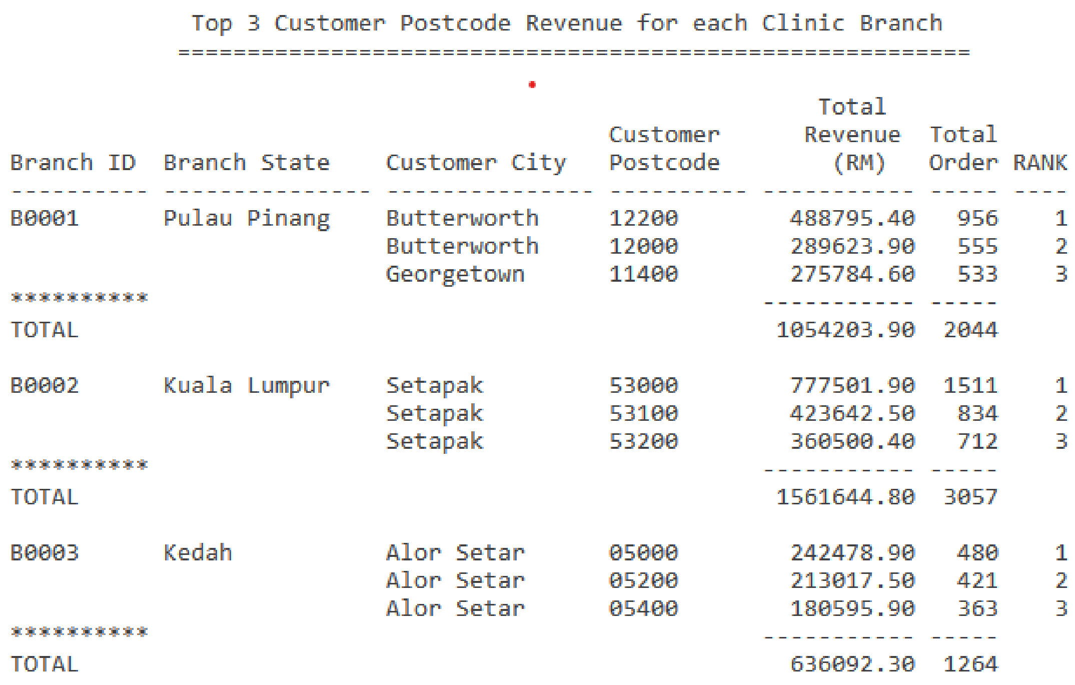
4.4.1 Query 1: Top Three Pets Owner's Postcode for each Branch based on total transaction amount (Strategic)

Purpose: The purpose of this query is to find out where the clinic's pet's owners are mostly from each branch. This is able to help top management level staff or the founder of this clinic to make the decision of opening a new branch or moving their current branch in the future which is where most of their owners come from. The query ranks the top three owners's postcodes based on the transaction amount from each pet's owner postcode.

SQL statement:

```
set pagesize 200
set linesize 80
clear break
clear compute
break on state on branch_id skip 1
COMPUTE SUM LABEL TOTAL OF TOTALSALES TOTAL_ORDER on
branch_id
TTITLE ON
TTITLE CENTER  'Top 3 Customer Postcode Revenue for each
Clinic Branch' SKIP 1-
CENTER
========================================================
= SKIP 2

column branch_id FORMAT a10
column branch_id HEADING 'Branch ID'
column STATE FORMAT a15
column STATE HEADING 'Branch State'
column CITY FORMAT a15
column CITY HEADING 'Customer City'
column POSTCODE FORMAT a10
column TotalSales format 9999999.99
column TotalSales HEADING 'Total  |Revenue |(RM)  '
column TotalSales format 9999999.99
column TOTAL_ORDER HEADING 'Total|Order'
column TOTAL_ORDER format 9999
column RANK format 99

SELECT branch_id,
       state,
       city,
       postcode,
       TotalSales, Total_Order, Rank
FROM
  ( SELECT T.branch_id,O.state,
           O.city,
           O.postcode,
           sum(Total_Amount) as TotalSales,
           count(distinct transaction_id) as
Total_Order,
           ROW_NUMBER() OVER (PARTITION BY O.state
                              ORDER BY sum(Total_Amount)
                               DESC)AS Rank
    FROM Transaction T, PetOwner O
    Where T.owner_id = O.owner_id
    Group by T.branch_id,O.state,O.city,O.postcode
  )
WHERE Rank <= 3
ORDER BY branch_id,TotalSales DESC;
```

Sample Output:



```
             Top 3 Customer Postcode Revenue for each Clinic Branch
            =========================================================

                                                           Total
                                               Customer   Revenue  Total
Branch ID  Branch State    Customer City   Postcode      (RM)  Order RANK
---------- --------------- --------------- ---------- ----------- ----- ----
B0001      Pulau Pinang    Butterworth     12200        488795.40   956    1
                           Butterworth     12000        289623.90   555    2
                           Georgetown      11400        275784.60   533    3
**********                                            ----------- -----
TOTAL                                                  1054203.90  2044

B0002      Kuala Lumpur    Setapak         53000        777501.90  1511    1
                           Setapak         53100        423642.50   834    2
                           Setapak         53200        360500.40   712    3
**********                                            ----------- -----
TOTAL                                                  1561644.80  3057

B0003      Kedah           Alor Setar      05000        242478.90   480    1
                           Alor Setar      05200        213017.50   421    2
                           Alor Setar      05400        180595.90   363    3
**********                                            ----------- -----
TOTAL                                                   636092.30  1264
```

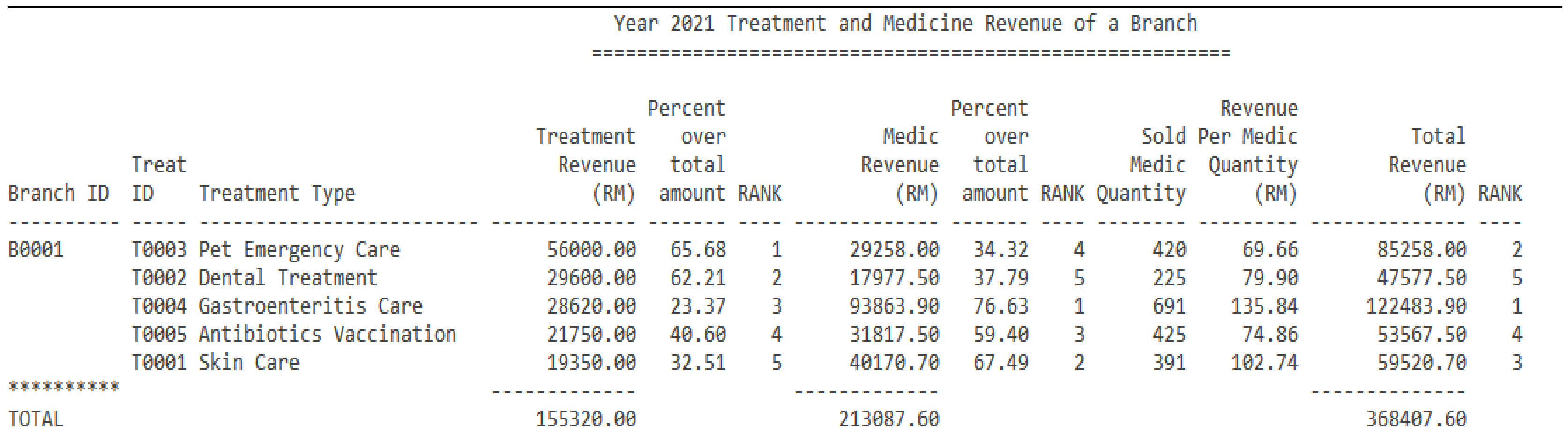
4.4.2 Query 2: Rank top treatment, to medicine sales and total revenue of a branch ()

Purpose: The purpose of this query is to rank the top treatment of a branch. This allows the branch clinic head or manager to know where the income from which type of treatments is. This query also shows that medicine revenue from each type of treatment. From this query, the branch head or manager can promote those treatments that have lower revenue.

SQL statement:
```
set pagesize 200
set linesize 160

clear break
clear compute
break on state on branch_id skip 1
COMPUTE SUM LABEL TOTAL OF Treatment_Revenue
Medic_Revenue Total_amount on branch_id
TTITLE ON
TTITLE CENTER  'Year 2021 Treatment and Medicine Revenue
of a Branch' SKIP 1-
CENTER
=======================================================
= SKIP 2

column TREATMENT_TYPE HEADING 'Treatment Type'
column TREATMENT_TYPE FORMAT A25
column TREATMENT_REVENUE HEADING
'Treatment|Revenue|(RM)'
column TREATMENT_REVENUE FORMAT 999999999.99
column MEDIC_REVENUE HEADING 'Medic|Revenue|(RM)'
column MEDIC_REVENUE FORMAT 999999999.99
column SOLD_QUANTITY HEADING 'Sold|Medic|Quantity'
column SOLD_QUANTITY FORMAT 99999
column Revenue_Per_Quantity HEADING 'Revenue|Per
Medic|Quantity|(RM)'
column Revenue_Per_Quantity FORMAT 99999.99
column Total_Amount HEADING 'Total|Revenue|(RM)'
column Total_Amount FORMAT 9999999999.99
column treatment_id HEADING 'Treatment|ID'
column Percent HEADING 'Percent|over|total|amount'
column Percent FORMAT 99.99


CREATE OR REPLACE VIEW FullRevenueGroupByTreatment2021
AS
select  T.branch_id,TT.treatment_id, TT.treatment_type,
sum(T.total_amount) as Total_Amount
from transaction T, appointment A, treatment TT
```

```
where A.appointment_id = T.appointment_id AND
A.treatment_id = TT.treatment_id AND EXTRACT(YEAR FROM
T.transaction_datetime) = '2021'
group by T.branch_id,TT.treatment_id, TT.treatment_type
order by branch_id;

CREATE OR REPLACE VIEW MedicRevenueGroupByTreatment2021
AS
select T.branch_id,TT.treatment_id, TT.treatment_type,
sum(TD.line_total) AS Medic_Revenue, sum(TD.line_qty) AS
Sold_Quantity, (sum(TD.line_total)/sum(TD.line_qty)) AS
Revenue_Per_Quantity
from transactiondetail TD, transaction T, appointment A,
treatment TT
where TD.transaction_id = T.transaction_id AND
A.appointment_id = T.appointment_id AND A.treatment_id =
TT.treatment_id AND EXTRACT(YEAR FROM
T.transaction_datetime) = '2021'
group by T.branch_id,TT.treatment_id, TT.treatment_type
order by branch_id;

select F.branch_id,F.treatment_id,F.treatment_type,
(F.Total_Amount - M.Medic_Revenue) AS Treatment_Revenue,
       ((F.Total_Amount - M.Medic_Revenue) /
F.Total_Amount * 100) AS Percent,
       ROW_NUMBER() OVER (ORDER BY (F.Total_Amount -
M.Medic_Revenue) DESC)AS Rank,
       M.Medic_Revenue,
       (M.Medic_Revenue / F.Total_Amount * 100) AS
Percent,
       ROW_NUMBER() OVER (ORDER BY M.Medic_Revenue
DESC)AS Rank,
       Sold_Quantity,Revenue_Per_Quantity,
F.Total_amount,
       ROW_NUMBER() OVER (ORDER BY F.Total_amount
DESC)AS Rank
from FullRevenueGroupByTreatment2021 F,
MedicRevenueGroupByTreatment2021 M
where F.branch_id = '&branch_id' AND F.branch_id =
M.branch_id AND F.treatment_id = M.treatment_id;
```

Sample Output:

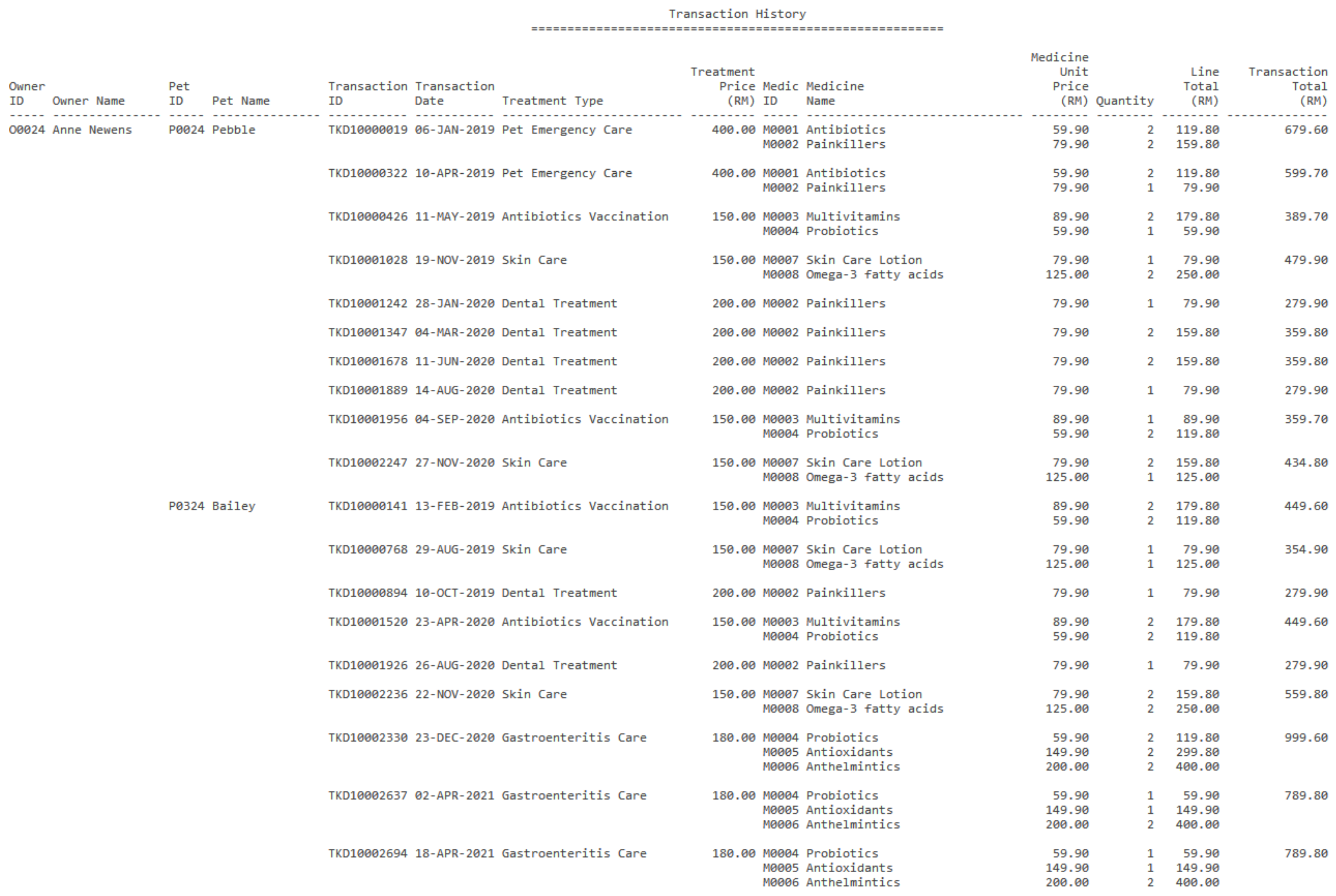


```
                                                      Year 2021 Treatment and Medicine Revenue of a Branch
                                                    =========================================================

                                                         Percent                    Percent                 Revenue
                                               Treatment    over              Medic    over          Sold Per Medic          Total
           Treat                                 Revenue   total            Revenue   total         Medic  Quantity        Revenue
Branch ID  ID    Treatment Type                     (RM)  amount RANK          (RM)  amount RANK Quantity      (RM)           (RM) RANK
---------- ----- ------------------------- ------------- ------- ---- ------------- ------- ---- -------- --------- -------------- ----
B0001      T0003 Pet Emergency Care             56000.00   65.68    1      29258.00   34.32    4      420     69.66       85258.00    2
           T0002 Dental Treatment               29600.00   62.21    2      17977.50   37.79    5      225     79.90       47577.50    5
           T0004 Gastroenteritis Care           28620.00   23.37    3      93863.90   76.63    1      691    135.84      122483.90    1
           T0005 Antibiotics Vaccination        21750.00   40.60    4      31817.50   59.40    3      425     74.86       53567.50    4
           T0001 Skin Care                      19350.00   32.51    5      40170.70   67.49    2      391    102.74       59520.70    3
**********                                 -------------                 -------------                                 --------------
TOTAL                                          155320.00                     213087.60                                      368407.60
```

4.4.3 Query 3: View Customer Transaction History by entering Customer ID (Operational)

Purpose: The purpose of this query is to check the previous transaction history by entering the owner ID. This clinic veterinarian checks the transaction history to view all treatments for all the owner's pets or what medic they have bought previously for their pet. All medical history of the owner's pets will also be shown.

```
clear break
clear compute
transaction history of a customer

set pagesize 200
set linesize 200
alter session set nls_date_format = 'DD-MON-YYYY';

TTITLE ON
TTITLE CENTER  'Transaction History' SKIP 1-
CENTER
=========================================================
= SKIP 2
break on owner_id on owner_name on pet_id on pet_name on
transaction_datetime on transaction_id on total_amount
on treatment_price on treatment_type skip 1
column OWNER_ID HEADING 'Owner|ID'
column OWNER_NAME HEADING 'Owner Name'
column OWNER_NAME FORMAT A15
column PET_ID HEADING 'Pet|ID'
column PET_NAME HEADING 'Pet Name'
column PET_NAME FORMAT A15
column TRANSACTION_ID HEADING 'Transaction|ID'
column TRANSACTION_ID FORMAT A11
column TRANSACTION_DATETIME HEADING 'Transaction|Date'
column TREATMENT_TYPE HEADING 'Treatment Type'
column TREATMENT_TYPE FORMAT A25
column TREATMENT_PRICE HEADING 'Treatment|Price|(RM)'
column TREATMENT_PRICE FORMAT 999.99
column MEDIC_ID HEADING 'Medic|ID'
column MEDIC_NAME HEADING 'Medicine|Name'
column MEDIC_PRICE HEADING 'Medicine|Unit|Price|(RM)'
column MEDIC_PRICE FORMAT 999.99
column LINE_QTY HEADING 'Quantity'
column LINE_QTY FORMAT 99
column LINE_TOTAL HEADING 'Line|Total|(RM)'
column LINE_TOTAL FORMAT 9999.99
column Total_Amount HEADING 'Transaction|Total|(RM)'
column Total_Amount FORMAT 9999999999.99

select
T.owner_id,PO.owner_name,P.pet_id,P.pet_name,T.transacti
on_id,
       T.transaction_datetime,
       TT.treatment_type, TT.treatment_price,
M.medic_id,M.medic_name,
       M.medic_price, TD.line_qty,
TD.line_total,T.total_amount
from medicalsupply M, transactiondetail TD, transaction
T, appointment A, treatment TT,PetOwner PO, Pet P
where M.medic_id = TD.medic_id AND TD.transaction_id =
T.transaction_id AND T.appointment_id = A.appointment_id
      AND A.treatment_id = TT.treatment_id AND A.pet_id
= P.pet_id
      AND PO.owner_id = T.Owner_id AND PO.Owner_id =
'&Owner_id'
order by pet_id,transaction_datetime, medic_id;
```



```
                                                                                             Transaction History
                                                                                 ========================================================

                                                                                                                                           Medicine
                                                                                            Treatment                                          Unit                Line    Transaction
Owner                Pet                  Transaction Transaction                               Price Medic Medicine                          Price                Total          Total
ID    Owner Name     ID    Pet Name       ID          Date        Treatment Type                 (RM) ID    Name                               (RM) Quantity     (RM)           (RM)
----- -------------- ----- -------------- ----------- ----------- ------------------------- --------- ----- ------------------------------ -------- -------- -------- --------------
O0024 Anne Newens    P0024 Pebble         TKD10000019 06-JAN-2019 Pet Emergency Care           400.00 M0001 Antibiotics                       59.90        2   119.80         679.60
                                                                                                      M0002 Painkillers                       79.90        2   159.80

                                          TKD10000322 10-APR-2019 Pet Emergency Care           400.00 M0001 Antibiotics                       59.90        2   119.80         599.70
                                                                                                      M0002 Painkillers                       79.90        1    79.90

                                          TKD10000426 11-MAY-2019 Antibiotics Vaccination      150.00 M0003 Multivitamins                     89.90        2   179.80         389.70
                                                                                                      M0004 Probiotics                        59.90        1    59.90

                                          TKD10001028 19-NOV-2019 Skin Care                    150.00 M0007 Skin Care Lotion                  79.90        1    79.90         479.90
                                                                                                      M0008 Omega-3 fatty acids              125.00        2   250.00

                                          TKD10001242 28-JAN-2020 Dental Treatment             200.00 M0002 Painkillers                       79.90        1    79.90         279.90

                                          TKD10001347 04-MAR-2020 Dental Treatment             200.00 M0002 Painkillers                       79.90        2   159.80         359.80

                                          TKD10001678 11-JUN-2020 Dental Treatment             200.00 M0002 Painkillers                       79.90        2   159.80         359.80

                                          TKD10001889 14-AUG-2020 Dental Treatment             200.00 M0002 Painkillers                       79.90        1    79.90         279.90

                                          TKD10001956 04-SEP-2020 Antibiotics Vaccination      150.00 M0003 Multivitamins                     89.90        1    89.90         359.70
                                                                                                      M0004 Probiotics                        59.90        2   119.80

                                          TKD10002247 27-NOV-2020 Skin Care                    150.00 M0007 Skin Care Lotion                  79.90        2   159.80         434.80
                                                                                                      M0008 Omega-3 fatty acids              125.00        1   125.00

                     P0324 Bailey         TKD10000141 13-FEB-2019 Antibiotics Vaccination      150.00 M0003 Multivitamins                     89.90        2   179.80         449.60
                                                                                                      M0004 Probiotics                        59.90        2   119.80

                                          TKD10000768 29-AUG-2019 Skin Care                    150.00 M0007 Skin Care Lotion                  79.90        1    79.90         354.90
                                                                                                      M0008 Omega-3 fatty acids              125.00        1   125.00

                                          TKD10000894 10-OCT-2019 Dental Treatment             200.00 M0002 Painkillers                       79.90        1    79.90         279.90

                                          TKD10001520 23-APR-2020 Antibiotics Vaccination      150.00 M0003 Multivitamins                     89.90        2   179.80         449.60
                                                                                                      M0004 Probiotics                        59.90        2   119.80

                                          TKD10001926 26-AUG-2020 Dental Treatment             200.00 M0002 Painkillers                       79.90        1    79.90         279.90

                                          TKD10002236 22-NOV-2020 Skin Care                    150.00 M0007 Skin Care Lotion                  79.90        2   159.80         559.80
                                                                                                      M0008 Omega-3 fatty acids              125.00        2   250.00

                                          TKD10002330 23-DEC-2020 Gastroenteritis Care         180.00 M0004 Probiotics                        59.90        2   119.80         999.60
                                                                                                      M0005 Antioxidants                     149.90        2   299.80
                                                                                                      M0006 Anthelmintics                    200.00        2   400.00

                                          TKD10002637 02-APR-2021 Gastroenteritis Care         180.00 M0004 Probiotics                        59.90        1    59.90         789.80
                                                                                                      M0005 Antioxidants                     149.90        1   149.90
                                                                                                      M0006 Anthelmintics                    200.00        2   400.00

                                          TKD10002694 18-APR-2021 Gastroenteritis Care         180.00 M0004 Probiotics                        59.90        1    59.90         789.80
                                                                                                      M0005 Antioxidants                     149.90        1   149.90
                                                                                                      M0006 Anthelmintics                    200.00        2   400.00
```

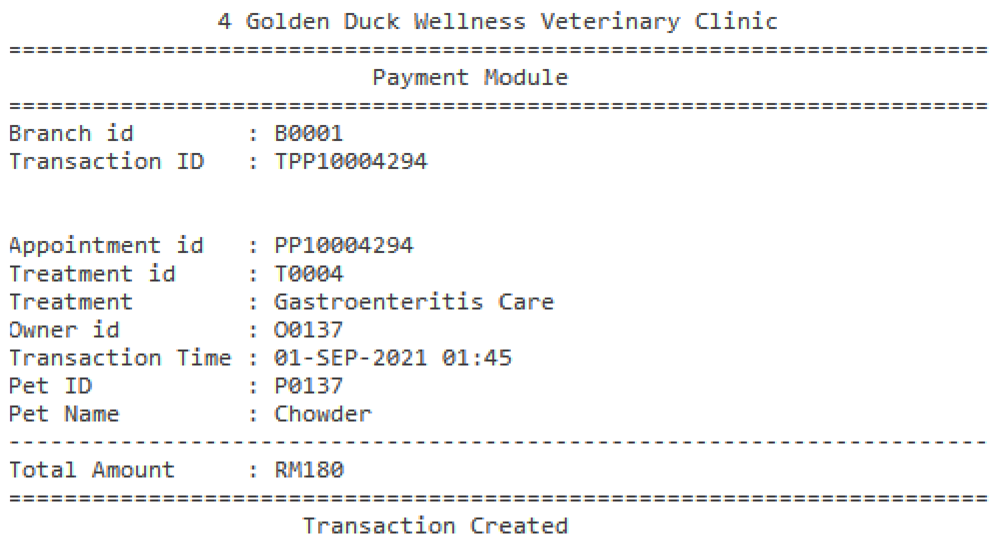
```
P0524 Mckenzie        TKD10000849 25-SEP-2019 Dental Treatment           200.00 M0002 Painkillers                      79.90         2   159.80          359.80

                      TKD10000873 04-OCT-2019 Pet Emergency Care         400.00 M0001 Antibiotics                      59.90         1    59.90          539.80
                                                                                M0002 Painkillers                      79.90         1    79.90

                      TKD10001585 12-MAY-2020 Dental Treatment           200.00 M0002 Painkillers                      79.90         2   159.80          359.80

                      TKD10001628 25-MAY-2020 Gastroenteritis Care       180.00 M0004 Probiotics                       59.90         1    59.90          789.80
                                                                                M0005 Antioxidants                    149.90         1   149.90
                                                                                M0006 Anthelmintics                   200.00         2   400.00

                      TKD10002677 12-APR-2021 Antibiotics Vaccination    150.00 M0003 Multivitamins                    89.90         2   179.80          449.60
                                                                                M0004 Probiotics                       59.90         2   119.80
```

4.4.4 Procedure 1: Create Transaction

Purpose: The purpose of this procedure is to allow the staff or veterinarian to create a transaction after the appointment and treatment of the owner's pet. The procedure will require the appointment id and will auto insert the owner id, branch id, update the total amount of the transition by adding the treatment price, and record the transaction date time. If any appointment id that entered is not found from the appointment table, an exception will be raised. Only existing appointments were able to make payment. The payment must be made after the appointment if not a trigger will be triggered and raise an exception. If the transaction of appointment has been made, an exception will be raised to notify the veterinarian that the transaction for this appointment has been created.

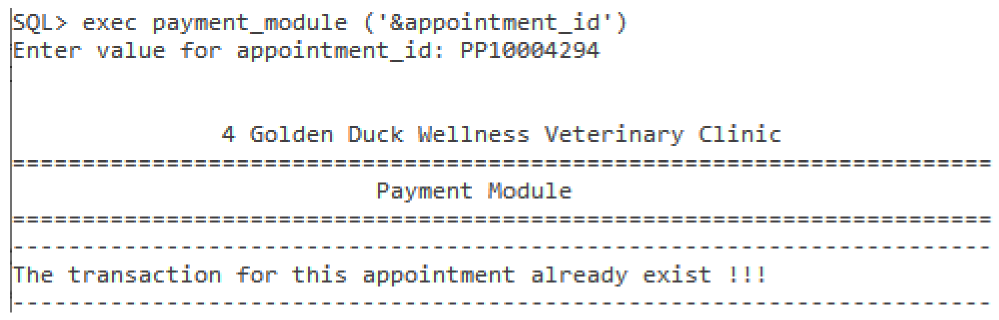
Procedure code:
```
SET SERVEROUTPUT ON FORMAT WRAPPED
alter session set nls_date_format = 'DD-MON-YYYY
HH24:MI';

CREATE OR REPLACE PROCEDURE payment_module
(IN_appointment_id in APPOINTMENT.appointment_id%TYPE)
IS

  appointmentCount        NUMBER :=0;
  transactionCount        NUMBER :=0;
  v_transaction_id      TRANSACTION.transaction_id%TYPE;
  v_treatment_id            TREATMENT.treatment_id%TYPE;
  v_treatment_price      TREATMENT.treatment_price%TYPE;
  v_owner_id
                              TRANSACTION.owner_id%TYPE;
  v_pet_id
                                        PET.pet_id%TYPE;
  v_branch_id                TRANSACTION.branch_id%TYPE;
  v_datetime
TRANSACTION.transaction_datetime%TYPE;
  v_appointment_time
APPOINTMENT.appointment_datetime%TYPE;
  v_totalamount           TRANSACTION.total_amount%TYPE;
  v_treatmenttype         TREATMENT.treatment_type%TYPE;
  v_pet_name
                                      PET.pet_name%TYPE;
  v_vet_id                APPOINTMENT.vet_id%TYPE;
  e_invalid_appointmentid EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_invalid_appointmentid,
-20150);
  e_repeated_transaction EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_repeated_transaction, -20151);

BEGIN
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(LPAD('4 Golden Duck Wellness
Veterinary Clinic',55, ' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',70,'='));
  DBMS_OUTPUT.PUT_LINE(LPAD('Payment Module',40, ' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',70,'='));

  select count(appointment_id) INTO appointmentCount
  from appointment
  where appointment_id = IN_appointment_id;

  IF appointmentCount = 0 THEN
         RAISE_APPLICATION_ERROR(-20150, 'Appointment ID
not Found !!!', true);
      END IF;

  select count(transaction_id) INTO transactionCount
  from transaction
  where appointment_id = IN_appointment_id;

  IF transactionCount > 0 THEN
         RAISE_APPLICATION_ERROR(-20151, 'Repeated
Transaction !!!', true);
      END IF;


  select treatment_id, pet_id, appointment_datetime,
vet_id
   INTO v_treatment_id, v_pet_id, v_appointment_time ,
v_vet_id
  FROM appointment
  WHERE appointment_id = IN_appointment_id;

  select treatment_price, treatment_type
    INTO v_treatment_price,v_treatmenttype
  FROM treatment
  WHERE treatment_id = v_treatment_id;

  select owner_id, pet_name
    INTO v_owner_id,v_pet_name
  FROM pet
  WHERE pet_id = v_pet_id;

  select branch_id
    INTO v_branch_id
  from veterinarian
  WHERE vet_id = v_vet_id;

  select sysdate
   into v_datetime
  from dual;
  v_totalamount := v_treatment_price;
  v_transaction_id := TO_CHAR('T'||IN_appointment_id);

  DBMS_OUTPUT.PUT_LINE('Branch id        : ' ||
v_Branch_id);
  DBMS_OUTPUT.PUT_LINE('Transaction ID   :
'||v_transaction_id);
  DBMS_OUTPUT.PUT_LINE(chr(10));


  DBMS_OUTPUT.PUT_LINE('Appointment id   : ' ||
IN_appointment_id);
  DBMS_OUTPUT.PUT_LINE('Treatment id     : ' ||
v_treatment_id);
  DBMS_OUTPUT.PUT_LINE('Treatment        : ' ||
v_treatmenttype);
  DBMS_OUTPUT.PUT_LINE('Owner id         : ' ||
v_owner_id);
  DBMS_OUTPUT.PUT_LINE('Transaction Time : ' ||
TO_CHAR(v_datetime));
  DBMS_OUTPUT.PUT_LINE('Pet ID           : ' ||
v_pet_id);
  DBMS_OUTPUT.PUT_LINE('Pet Name         : ' ||
v_pet_name);
  DBMS_OUTPUT.PUT_LINE(LPAD('-',70,'-'));
  DBMS_OUTPUT.PUT_LINE('Total Amount     : RM' ||
to_char (v_totalamount, '9999.99'));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',70,'='));
  DBMS_OUTPUT.PUT_LINE(LPAD('Transaction Created',40, '
'));
  Insert into transaction
values(v_transaction_id,v_owner_id,IN_appointment_id,v_B
ranch_id,v_totalamount,v_datetime);



  EXCEPTION
    WHEN e_invalid_appointmentid THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',70,'-'));
      DBMS_OUTPUT.PUT_LINE('Transaction Fail to Create
!!! Entered Appointment does not exist !!!');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',70,'-'));
    WHEN e_repeated_transaction THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',70,'-'));
      DBMS_OUTPUT.PUT_LINE('The transaction for this
appointment already exist !!!');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',70,'-'));

END;
/
exec payment_module ('&appointment_id')
```

Sample Output:



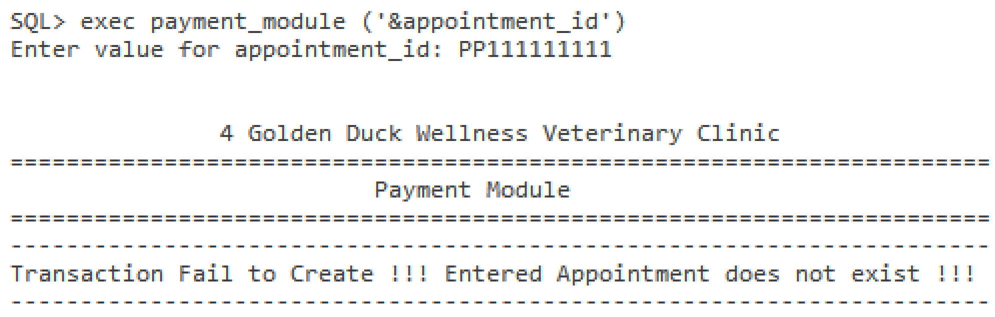
```
               4 Golden Duck Wellness Veterinary Clinic
======================================================================
                          Payment Module
======================================================================
Branch id         : B0001
Transaction ID    : TPP10004294


Appointment id    : PP10004294
Treatment id      : T0004
Treatment         : Gastroenteritis Care
Owner id          : O0137
Transaction Time  : 01-SEP-2021 01:45
Pet ID            : P0137
Pet Name          : Chowder
----------------------------------------------------------------------
Total Amount      : RM180
======================================================================
                     Transaction Created
```

Exception Output: Repeated appointment id



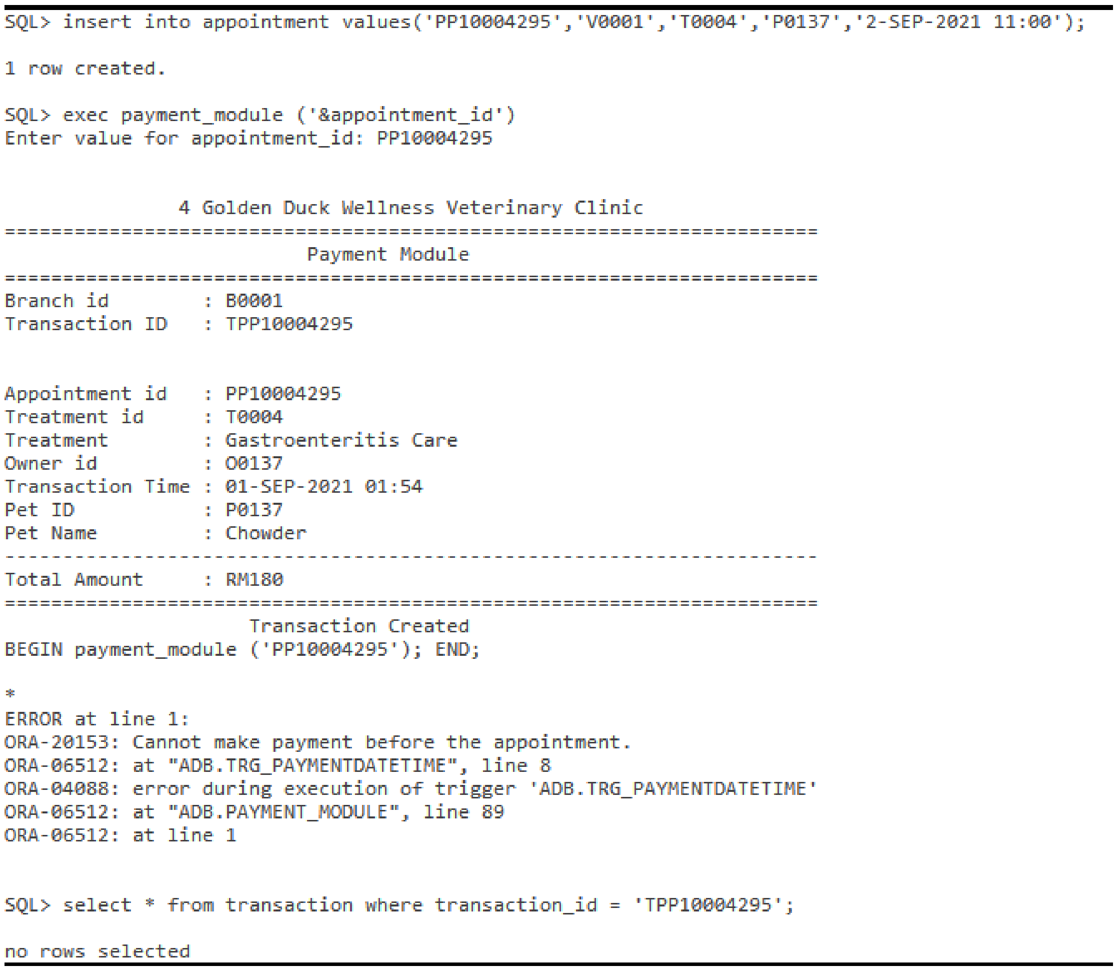
```
SQL> exec payment_module ('&appointment_id')
Enter value for appointment_id: PP10004294


               4 Golden Duck Wellness Veterinary Clinic
======================================================================
                          Payment Module
======================================================================
----------------------------------------------------------------------
The transaction for this appointment already exist !!!
----------------------------------------------------------------------
```

Exception Output: Invalid appointment ID



```
SQL> exec payment_module ('&appointment_id')
Enter value for appointment_id: PP111111111


               4 Golden Duck Wellness Veterinary Clinic
======================================================================
                          Payment Module
======================================================================
----------------------------------------------------------------------
Transaction Fail to Create !!! Entered Appointment does not exist !!!
----------------------------------------------------------------------
```

Exception Output: Triggered when make payment before appointment date time



```
SQL> insert into appointment values('PP10004295','V0001','T0004','P0137','2-SEP-2021 11:00');

1 row created.

SQL> exec payment_module ('&appointment_id')
Enter value for appointment_id: PP10004295


              4 Golden Duck Wellness Veterinary Clinic
======================================================================
                          Payment Module
======================================================================
Branch id        : B0001
Transaction ID   : TPP10004295


Appointment id   : PP10004295
Treatment id     : T0004
Treatment        : Gastroenteritis Care
Owner id         : O0137
Transaction Time : 01-SEP-2021 01:54
Pet ID           : P0137
Pet Name         : Chowder
----------------------------------------------------------------------
Total Amount     : RM180
======================================================================
                     Transaction Created
BEGIN payment_module ('PP10004295'); END;

*
ERROR at line 1:
ORA-20153: Cannot make payment before the appointment.
ORA-06512: at "ADB.TRG_PAYMENTDATETIME", line 8
ORA-04088: error during execution of trigger 'ADB.TRG_PAYMENTDATETIME'
ORA-06512: at "ADB.PAYMENT_MODULE", line 89
ORA-06512: at line 1


SQL> select * from transaction where transaction_id = 'TPP10004295';

no rows selected
```

4.4.5 Procedure 2: Add transaction detail

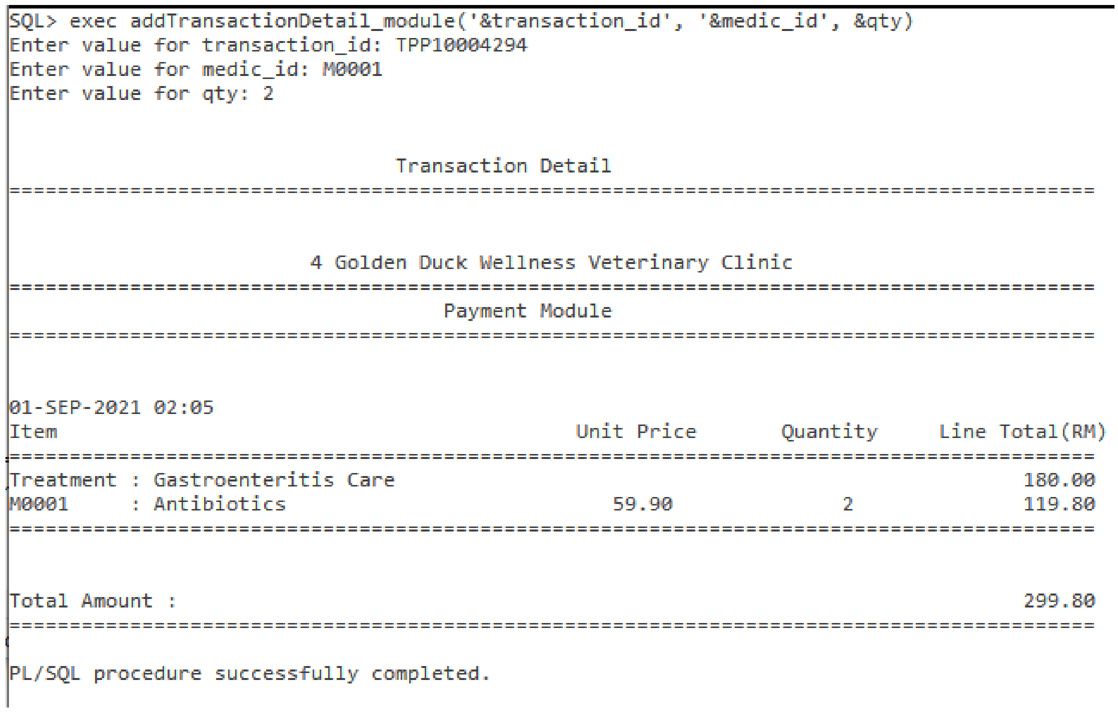
Purpose: The purpose of this procedure is to allow the staff or veterinarian to add transaction details such as medicine bought by the pet's owner after creating transaction details. For example, adding medicine with id 'M0001' with quantity to a transaction. After adding transaction detail a trigger will be triggered to automatically update the total amount of the transaction. This procedure consists of validating whether the transaction id exists, the quantity must be more than zero, whether there is repeated medicine in the transaction.

Procedure code:

```
SET SERVEROUTPUT ON FORMAT WRAPPED
alter session set nls_date_format = 'DD-MON-YYYY
HH24:MI';

CREATE OR REPLACE PROCEDURE addTransactionDetail_module
(in_transaction_id in TRANSACTION.transaction_id%TYPE,
in_medic_id in TRANSACTIONDETAIL.transaction_id%TYPE,
in_qty in TRANSACTIONDETAIL.line_qty%TYPE) IS

  transactionCount        NUMBER :=0;
  medicCount              NUMBER :=0;
  v_treatmenttype         TREATMENT.treatment_type%TYPE;
  v_treatment_price      TREATMENT.treatment_price%TYPE;
  v_medic_price          MEDICALSUPPLY.medic_price%TYPE;
  v_medic_price2         MEDICALSUPPLY.medic_price%TYPE;
  v_medic_name            MEDICALSUPPLY.medic_name%TYPE;
  v_medic_name2           MEDICALSUPPLY.medic_name%TYPE;
  v_line_total
                      TRANSACTIONDETAIL.line_total%TYPE;
  v_transaction_id
TRANSACTIONDETAIL.transaction_id%TYPE;
  v_totalamount           TRANSACTION.total_amount%TYPE;
  v_transactiondate
TRANSACTION.transaction_datetime%TYPE;
  v_daydifferent          number;
  e_invalid_transactionid EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_invalid_transactionid,
-20154);
  e_repeated_medicid EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_repeated_medicid, -20155);
  e_zero_qty EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_zero_qty, -20156);
  e_dayexceed EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_dayexceed, -20157);


  CURSOR detail_cursor IS
     SELECT *
     FROM transactiondetail
     WHERE transaction_id = in_transaction_id;

BEGIN
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(LPAD('Transaction Detail',50, '
'));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));

  select count(transaction_id)
   INTO transactionCount
  FROM transaction
  WHERE transaction_id = in_transaction_id;

  IF transactionCount = 0 THEN
         RAISE_APPLICATION_ERROR(-20154, 'Transaction ID
not Found and Not Created yet !!!', true);
      END IF;

  SELECT SYSDATE - transaction_datetime,
transaction_datetime INTO
v_daydifferent,v_transactiondate
  FROM DUAL, transaction
  where transaction_id = in_transaction_id ;

  IF v_daydifferent > 7 THEN
         RAISE_APPLICATION_ERROR(-20157, 'Day Exceed',
true);
      END IF;

  IF in_qty <= 0 THEN
         RAISE_APPLICATION_ERROR(-20156, 'More than
zero', true);
      END IF;

  select count(transaction_id)
   INTO transactionCount
  FROM transaction
  WHERE transaction_id = in_transaction_id;

  IF transactionCount = 0 THEN
         RAISE_APPLICATION_ERROR(-20154, 'Transaction ID
not Found and Not Created yet !!!', true);
      END IF;

  select count(medic_id)
   INTO medicCount
  FROM transactionDetail
  WHERE medic_id = in_medic_id AND transaction_id =
in_transaction_id;

  IF medicCount >0 THEN
```

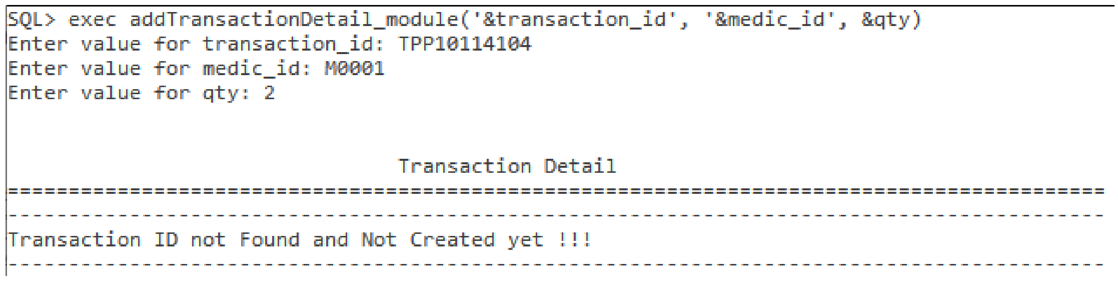
```
         RAISE_APPLICATION_ERROR(-20155, 'The Medic for
this transaction already existed !!!', true);
      END IF;

  select medic_price, medic_name
   INTO v_medic_price, v_medic_name
  FROM medicalsupply
  WHERE medic_id = in_medic_id;

  v_line_total := in_qty * v_medic_price;

  select treatment_type, treatment_price into
v_treatmenttype, v_treatment_price
  from transaction T, appointment A, Treatment R
  where T.transaction_id = in_transaction_id AND
T.appointment_id = A.appointment_id AND A.treatment_id =
R.treatment_id;

  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(LPAD('4 Golden Duck Wellness
Veterinary Clinic',65, ' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
  DBMS_OUTPUT.PUT_LINE(LPAD('Payment Module',50, ' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(v_transactiondate));
  DBMS_OUTPUT.PUT_LINE('Item'|| LPAD('Unit Price',53,'
') || LPAD('Quantity',15,' ')|| LPAD('Line
Total(RM)',19,' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
  DBMS_OUTPUT.PUT_LINE('Treatment : '||
RPAD(v_treatmenttype,35,' ') ||
LPAD(to_char(v_treatment_price, '9999.99'),'43',' '));

  IF transactionCount >0 THEN
    FOR detail_record IN detail_cursor LOOP
       select medic_name, medic_price into
v_medic_name2, v_medic_price2
       from medicalsupply
       where medic_id = detail_record.medic_id;
       DBMS_OUTPUT.PUT_LINE(detail_record.medic_id||'
: ' || RPAD(v_medic_name2,35,'
')||to_char(v_medic_price2,
'9999.99')||LPAD(detail_record.line_qty,15,'
')||LPAD(to_char(detail_record.line_total,
'9999.99'),20,' '));

    END LOOP;
   END IF;
  DBMS_OUTPUT.PUT_LINE(in_medic_id || '     : ' ||
RPAD(v_medic_name,35,' ')|| to_char(v_medic_price,
'9999.99')||LPAD(in_qty,15,'
')||LPAD(to_char(v_line_total, '9999.99'),20,' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));


  insert into transactiondetail
values(in_transaction_id,in_medic_id,in_qty,v_line_total
);
  select total_amount into v_totalamount
  from transaction where transaction_id =
in_transaction_id;
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE('Total Amount : '||
LPAD(to_char(v_totalamount, '9999.99'),'75',' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));



  EXCEPTION
    WHEN e_invalid_transactionid THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Transaction ID not Found
and Not Created yet !!!');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
    WHEN e_repeated_medicid THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('The Medic type already
exist if you wish edit run edit procedure');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
    WHEN e_zero_qty THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('The quantity must be more
than zero');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
    WHEN e_dayexceed THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Transaction that past 7
days cannot be edited');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));



END;
/
```

Sample Output: After adding the total amount will be updated



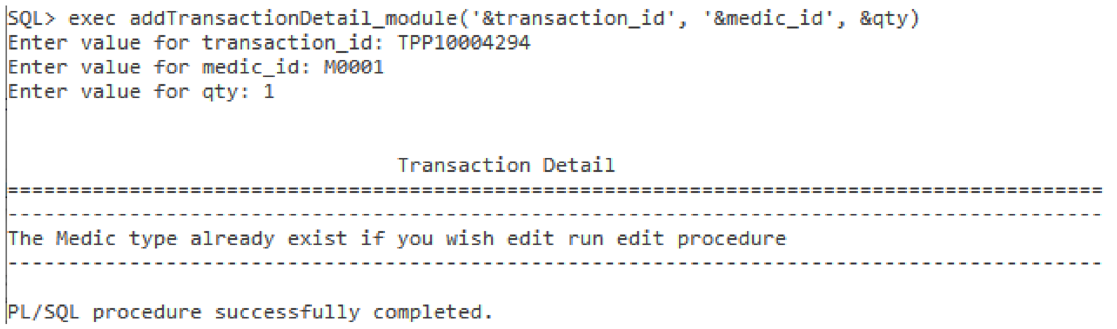
```
SQL> exec addTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10004294
Enter value for medic_id: M0001
Enter value for qty: 2


                                Transaction Detail
==========================================================================================


                          4 Golden Duck Wellness Veterinary Clinic
==========================================================================================
                                    Payment Module
==========================================================================================


01-SEP-2021 02:05
Item                                           Unit Price        Quantity     Line Total(RM)
==========================================================================================
Treatment : Gastroenteritis Care                                                    180.00
M0001     : Antibiotics                          59.90                2             119.80
==========================================================================================


Total Amount :                                                                      299.80
==========================================================================================

PL/SQL procedure successfully completed.
```

Exception output Output: When transaction id do not exist.



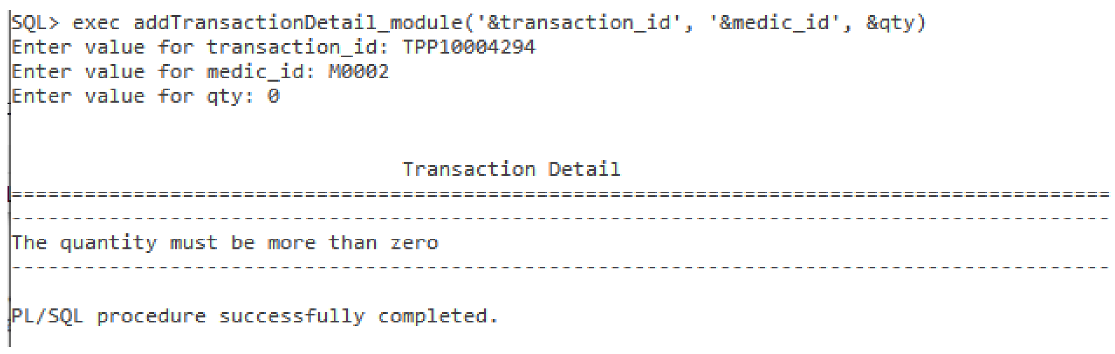
```
SQL> exec addTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10114104
Enter value for medic_id: M0001
Enter value for qty: 2


                                Transaction Detail
===========================================================================================
-------------------------------------------------------------------------------------------
Transaction ID not Found and Not Created yet !!!
-------------------------------------------------------------------------------------------
```

Exception output Output: When medic id entered it already exist in the transaction.



```
SQL> exec addTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10004294
Enter value for medic_id: M0001
Enter value for qty: 1


                                Transaction Detail
===========================================================================================
-------------------------------------------------------------------------------------------
The Medic type already exist if you wish edit run edit procedure
-------------------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

```
Exception output Output: When medic quantity less than 0.
```



```
SQL> exec addTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10004294
Enter value for medic_id: M0002
Enter value for qty: 0


                                 Transaction Detail
==========================================================================================
------------------------------------------------------------------------------------------
The quantity must be more than zero
------------------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

4.4.6 Procedure 3: Edit or delete transaction detail

Purpose: The purpose of this procedure is to allow the staff or veterinarian to edit or delete a transaction detail if any mistake entry is made. For example, extra medicine charges for the pet's owner, or require more quantity of a medicine. The procedure will validate that the quantity must more than or equal 0. If the quantity is 0 means delete that transaction detail. If the transaction is a past 7 days transaction, the transaction details are not allowed to be edited. If the transaction detail entered did not exist an error will be prompted and any edition or deletion will update the transaction total amount.

Procedure code:

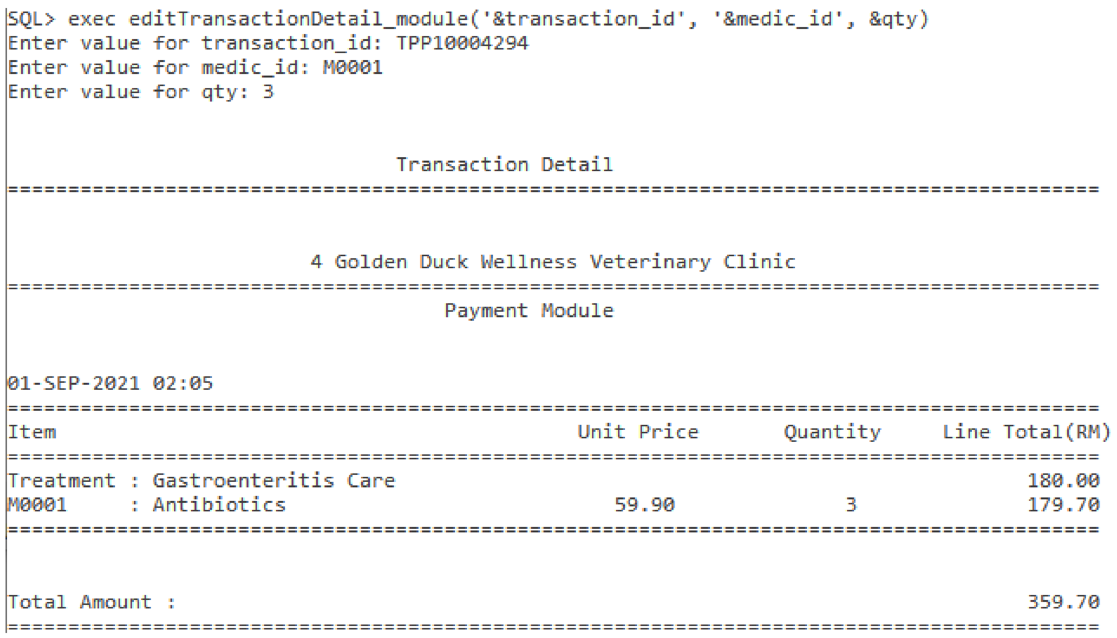
```
CREATE OR REPLACE PROCEDURE editTransactionDetail_module
(in_transaction_id in TRANSACTION.transaction_id%TYPE,
in_medic_id in TRANSACTIONDETAIL.medic_id%TYPE, in_qty
in TRANSACTIONDETAIL.line_qty%TYPE) IS

  transactionDetailCount        NUMBER :=0;
  medicCount                    NUMBER :=0;
  v_treatmenttype
TREATMENT.treatment_type%TYPE;
  v_treatment_price
TREATMENT.treatment_price%TYPE;
  v_medic_price          MEDICALSUPPLY.medic_price%TYPE;
  v_medic_name            MEDICALSUPPLY.medic_name%TYPE;
  v_new_line_total
TRANSACTIONDETAIL.line_total%TYPE;
  v_line_total
TRANSACTIONDETAIL.line_total%TYPE;
  v_line_qty
TRANSACTIONDETAIL.line_qty%TYPE;
  v_transaction_id
TRANSACTIONDETAIL.transaction_id%TYPE;
  v_transactiondate
TRANSACTION.transaction_datetime%TYPE;
  v_daydifferent                number;
  v_samequantity                number;
  v_totalamount
TRANSACTION.total_amount%TYPE;
  e_invalid_transactiondetail EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_invalid_transactiondetail,
-20158);
  e_lesszero_qty EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_lesszero_qty, -20159);
  e_day_exceed EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_day_exceed, -20160);
  e_samequantity EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_samequantity, -20161);



  CURSOR detail_cursor IS
     SELECT *
     FROM transactiondetail
     WHERE transaction_id = in_transaction_id;

BEGIN

  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(LPAD('Transaction Detail',50, '
'));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));

  select count(*)
   INTO transactionDetailCount
  FROM transactionDetail
  WHERE transaction_id = in_transaction_id AND medic_id
= in_medic_id ;

  IF transactionDetailCount = 0 THEN
         RAISE_APPLICATION_ERROR(-20158, 'Invalid
Transaction Detail', true);
      END IF;

  IF in_qty < 0 THEN
         RAISE_APPLICATION_ERROR(-20159, 'Cannot
Negative', true);
      END IF;

  SELECT SYSDATE - transaction_datetime,
transaction_datetime INTO
v_daydifferent,v_transactiondate
  FROM DUAL, transaction
  where transaction_id = in_transaction_id ;

  IF v_daydifferent > 7 THEN
         RAISE_APPLICATION_ERROR(-20160, 'Day Exceed',
true);
```

```
      END IF;



  select count(*)
   INTO v_samequantity
  FROM transactionDetail
  WHERE transaction_id = in_transaction_id AND medic_id
= in_medic_id AND line_qty = in_qty ;

  IF v_samequantity > 0 THEN
         RAISE_APPLICATION_ERROR(-20161, 'Same
quantity', true);
      END IF;

  select line_qty, line_total
   INTO v_line_qty, v_line_total
  FROM transactionDetail
  WHERE transaction_id = in_transaction_id AND medic_id
= in_medic_id ;

  select medic_price, medic_name
   INTO v_medic_price, v_medic_name
  FROM medicalsupply
  WHERE medic_id = in_medic_id;



  select treatment_type, treatment_price into
v_treatmenttype, v_treatment_price
  from transaction T, appointment A, Treatment R
  where T.transaction_id = in_transaction_id AND
T.appointment_id = A.appointment_id AND A.treatment_id =
R.treatment_id;

  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(LPAD('4 Golden Duck Wellness
Veterinary Clinic',65, ' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
  DBMS_OUTPUT.PUT_LINE(LPAD('Payment Module',50, ' '));
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE(TO_CHAR(v_transactiondate));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
  DBMS_OUTPUT.PUT_LINE('Item'|| LPAD('Unit Price',53,'
') || LPAD('Quantity',15,' ')|| LPAD('Line
Total(RM)',19,' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
  DBMS_OUTPUT.PUT_LINE('Treatment : '||
RPAD(v_treatmenttype,35,' ') ||
LPAD(to_char(v_treatment_price, '9999.99'),'43',' '));
```

```
    FOR detail_record IN detail_cursor LOOP
       IF detail_record.medic_id != in_medic_id THEN
        select medic_name, medic_price into
v_medic_name, v_medic_price
        from medicalsupply
        where medic_id = detail_record.medic_id;
       DBMS_OUTPUT.PUT_LINE(detail_record.medic_id||'
: ' || RPAD(v_medic_name,35,' ')||to_char(v_medic_price,
'9999.99')||LPAD(detail_record.line_qty,15,'
')||LPAD(to_char(detail_record.line_total,
'9999.99'),20,' '));
       END IF;

    END LOOP;


  IF in_qty = 0 THEN
         DELETE FROM transactiondetail
           WHERE transaction_id = in_transaction_id AND
medic_id = in_medic_id;

  ELSE
    select medic_name, medic_price into v_medic_name,
v_medic_price
        from medicalsupply
        where medic_id = in_medic_id;
    v_new_line_total := in_qty * v_medic_price;
    UPDATE TRANSACTIONDETAIL
      SET line_qty = in_qty, line_total =
v_new_line_total
           WHERE transaction_id = in_transaction_id AND
medic_id = in_medic_id;
    DBMS_OUTPUT.PUT_LINE(in_medic_id||'     : '
||RPAD(v_medic_name,35,' ')||to_char(v_medic_price,
'9999.99')||LPAD(in_qty,15,'
')||LPAD(to_char(v_new_line_total, '9999.99'),20,' '));
  END IF;

  select total_amount into v_totalamount
  from transaction where transaction_id =
in_transaction_id;
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
  DBMS_OUTPUT.PUT_LINE(chr(10));
  DBMS_OUTPUT.PUT_LINE('Total Amount : '||
LPAD(to_char(v_totalamount, '9999.99'),'75',' '));
  DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));



  EXCEPTION
```

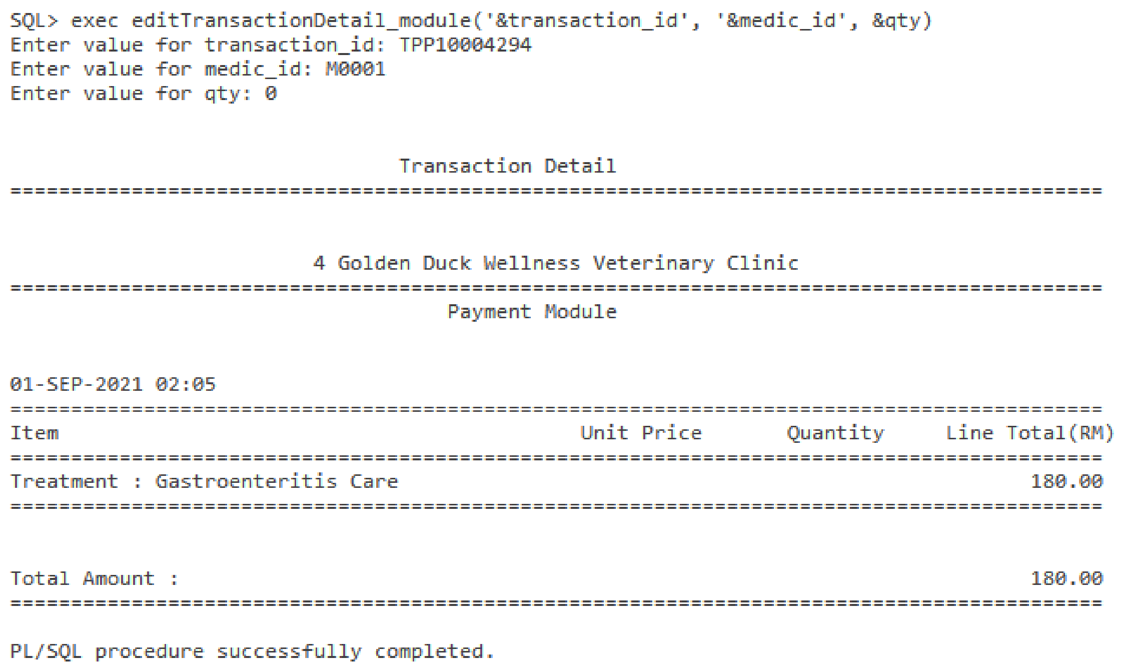
```
    WHEN e_invalid_transactiondetail THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Invalid Transaction Detail
and not FOUND !!!');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
    WHEN e_lesszero_qty THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Quantity Cannot Less Than 0
!!!');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
    WHEN e_day_exceed THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Transaction more than 7
days can be modify');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
    WHEN e_samequantity THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Same as previous quantity
no changes needed');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));

END;
/
```

Sample Output: After editing existing transaction details, the total amount will be automatically updated.



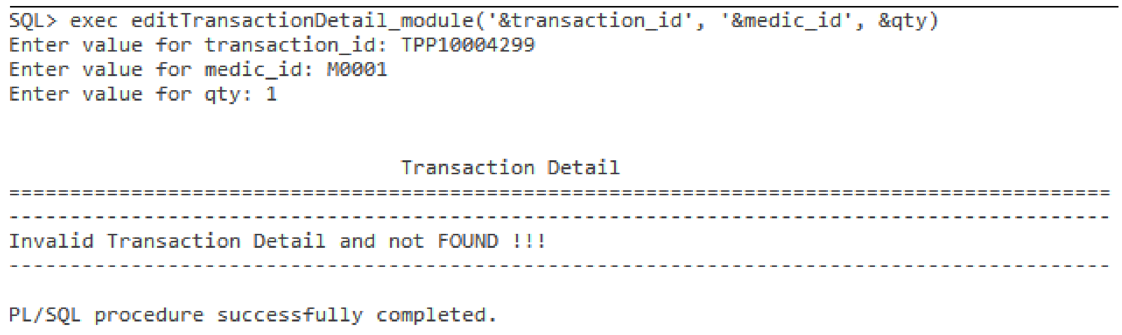
```
SQL> exec editTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10004294
Enter value for medic_id: M0001
Enter value for qty: 3


                                Transaction Detail
==========================================================================================


                        4 Golden Duck Wellness Veterinary Clinic
==========================================================================================
                                    Payment Module


01-SEP-2021 02:05
==========================================================================================
Item                                               Unit Price       Quantity     Line Total(RM)
==========================================================================================
Treatment : Gastroenteritis Care                                                        180.00
M0001     : Antibiotics                               59.90               3             179.70
==========================================================================================


Total Amount :                                                                          359.70
==========================================================================================
```

Sample Output: After deleting existing transaction details, the total amount will be automatically updated.



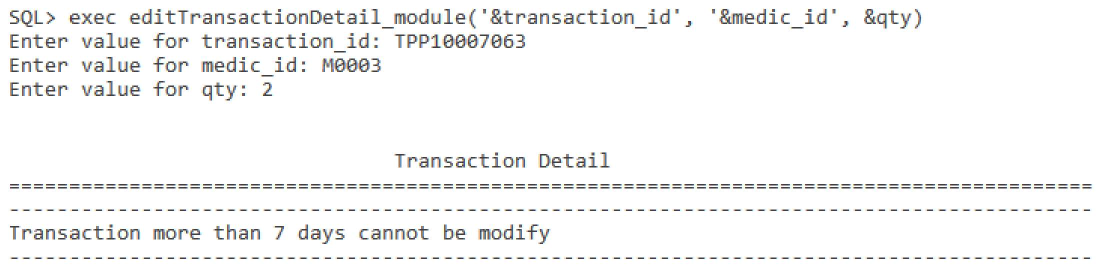
```
SQL> exec editTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10004294
Enter value for medic_id: M0001
Enter value for qty: 0


                                Transaction Detail
==========================================================================================


                         4 Golden Duck Wellness Veterinary Clinic
==========================================================================================
                                    Payment Module


01-SEP-2021 02:05
==========================================================================================
Item                                               Unit Price       Quantity      Line Total(RM)
==========================================================================================
Treatment : Gastroenteritis Care                                                      180.00
==========================================================================================


Total Amount :                                                                        180.00
==========================================================================================

PL/SQL procedure successfully completed.
```

Exception Output: Entered transaction details that do not exist, an error message will be displayed.



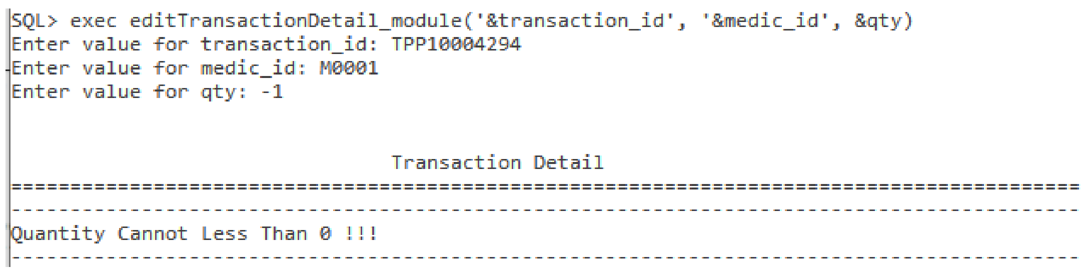
```
SQL> exec editTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10004299
Enter value for medic_id: M0001
Enter value for qty: 1


                                Transaction Detail
===========================================================================================
-------------------------------------------------------------------------------------------
Invalid Transaction Detail and not FOUND !!!
-------------------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

Exception Output: Past 7 days transaction cannot be edited.



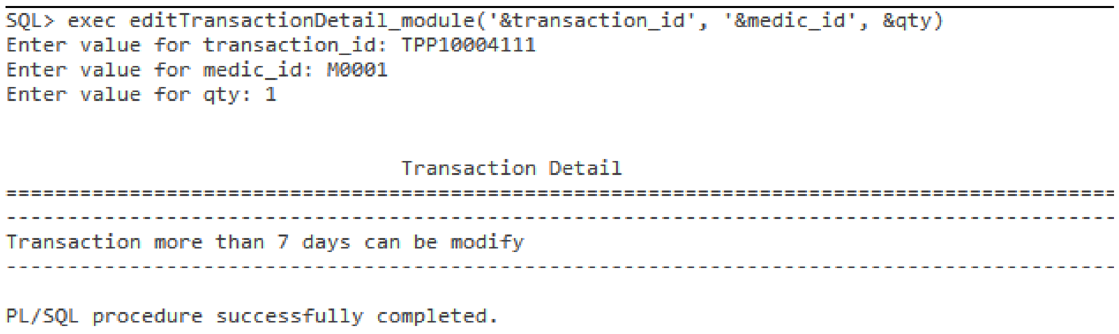
```
SQL> exec editTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10007063
Enter value for medic_id: M0003
Enter value for qty: 2


                                Transaction Detail
==========================================================================================
------------------------------------------------------------------------------------------
Transaction more than 7 days cannot be modify
------------------------------------------------------------------------------------------
```

Exception Output: Entered less than 0 , an error message will be displayed.



```
SQL> exec editTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10004294
Enter value for medic_id: M0001
Enter value for qty: -1


                                Transaction Detail
==========================================================================================
------------------------------------------------------------------------------------------
Quantity Cannot Less Than 0 !!!
------------------------------------------------------------------------------------------
```

Exception Output: Past 7 days transactions are not allowed to be edited.



```
SQL> exec editTransactionDetail_module('&transaction_id', '&medic_id', &qty)
Enter value for transaction_id: TPP10004111
Enter value for medic_id: M0001
Enter value for qty: 1


                                Transaction Detail
==========================================================================================
------------------------------------------------------------------------------------------
Transaction more than 7 days can be modify
------------------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

4.4.7 Trigger 1: Validate Payment Date Time

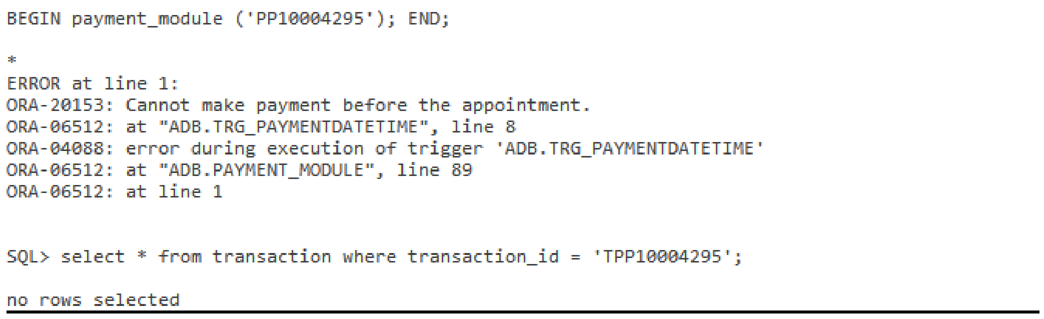
Purpose: The purpose of this trigger is to validate payment date time. The pet's owner is only able to make payment after the appointment. It is because the customer might cancel the appointment before the appointment date time. Thus, it only makes sense that the pet's owner pays after the treatment for his or her pet has been done.

Trigger code:

```
CREATE OR REPLACE TRIGGER trg_paymentDateTime
  BEFORE INSERT ON Transaction
  FOR EACH ROW
DECLARE
  v_appointment_date
APPOINTMENT.appointment_datetime%TYPE;
BEGIN
  Select appointment_datetime into v_appointment_date
  FROM appointment
  Where appointment_id = :new.appointment_id;
  IF :new.transaction_datetime<v_appointment_date THEN
    RAISE_APPLICATION_ERROR(-20153, 'Cannot make payment
before the appointment.' );
```

```
  END IF;
END;
/
```

Sample Output:



```
BEGIN payment_module ('PP10004295'); END;

*
ERROR at line 1:
ORA-20153: Cannot make payment before the appointment.
ORA-06512: at "ADB.TRG_PAYMENTDATETIME", line 8
ORA-04088: error during execution of trigger 'ADB.TRG_PAYMENTDATETIME'
ORA-06512: at "ADB.PAYMENT_MODULE", line 89
ORA-06512: at line 1


SQL> select * from transaction where transaction_id = 'TPP10004295';

no rows selected
```

4.4.8 Trigger 2: Update medicine quantity and transaction total amount after editing transaction detail

Purpose: The veterinarian might enter the transaction details wrongly. For example, the veterinarian charges the pet's owner 3 Skin Care Lotions but the pet's owner only buys 2 instead of 3. This allows the veterinarian to edit the quantity that is bought by the pet's owner. Same goes to if the pet's owner wishes to buy even after entering the transaction details. After editing, ths trigger will auto update the stock quantity and the total amount of the transaction. Before is use in this case because before editing the system needs to make sure the stock quantity is enough before selling to the pet's owner.

Trigger code:

```
CREATE OR REPLACE TRIGGER TRG_Update_EditTransaction
  BEFORE UPDATE ON TransactionDetail
  FOR EACH ROW
DECLARE
  v_quantitydiff    TRANSACTIONDETAIL.line_qty%TYPE;
  v_newqty          TRANSACTIONDETAIL.line_qty%TYPE;
  v_newlinetotal    TRANSACTIONDETAIL.line_total%TYPE;
  v_medicprice      MEDICALSUPPLY.medic_price%TYPE;
  v_linetotal_dif   TRANSACTIONDETAIL.line_total%TYPE;

 BEGIN
  Select medic_price into v_medicprice
  from medicalsupply where medic_id = :old.medic_id;

  v_newlinetotal := :new.line_total;

  IF :new.line_qty > :old.line_qty THEN
   Update medicalsupply
    SET medic_qty = medic_qty - (:new.line_qty -
:old.line_qty)
    where medic_id = :old.medic_id;
   v_linetotal_dif := v_newlinetotal - :old.line_total;
   Update Transaction
    SET total_amount = total_amount + v_linetotal_dif
    where transaction_id = :old.transaction_id;

  ELSE
   Update medicalsupply
    SET medic_qty = medic_qty + (:old.line_qty -
:new.line_qty )
    where medic_id = :old.medic_id;
   v_linetotal_dif := :old.line_total - v_newlinetotal;
   Update Transaction
    SET total_amount = total_amount - v_linetotal_dif
    where transaction_id = :old.transaction_id;
  END IF;

END;/
```

4.4.6 Trigger 3: Update the stock and transaction total amount after deleting transaction detail

Purpose: The purpose of this trigger is to Update the stock and transaction total amount after deleting transaction details . The veterinarian might accidentally charge the pet's owner with medicine that the pet owner didn't purchase. Hence, after the pet's owner or the veterinarian noticed the mistake. The veterinarian will delete that transaction detail. Thus this trigger will automatically the stock quantity and the total amount of the transaction.

Trigger code:
```
CREATE OR REPLACE TRIGGER TRG_Update_DeleteTransaction
  BEFORE DELETE ON TransactionDetail
  FOR EACH ROW
 BEGIN
  Update Transaction
   SET total_amount = total_amount - :old.line_total
   where transaction_id = :old.transaction_id;
  Update MedicalSupply
   SET medic_qty = medic_qty + :old.line_qty
   where medic_id = :old.medic_id;
END;
/
```

4.4.7 Trigger 4: Update transaction total amount

Purpose: The purpose of this trigger is to keep adding the transaction amount after the pet's owner keep buying new medicine for his or her pet after the treatment.

Trigger code:
```
CREATE OR REPLACE TRIGGER TRG_Update_Total_Amount
  After Insert ON TransactionDetail
  FOR EACH ROW
 BEGIN
  Update Transaction
   SET total_amount = total_amount + :new.line_total
   where transaction_id = :new.transaction_id;
END;
/
```

4.4.11 Report 1: Summary report of Veterinarian Performance in a month.

Purpose: The purpose of this report is to monitor the performance of a veterinarian in a month and how much revenue the veterinarian brings to the company in a specific month. The number of types of treatments that are handled by the veterinarian will also be shown.

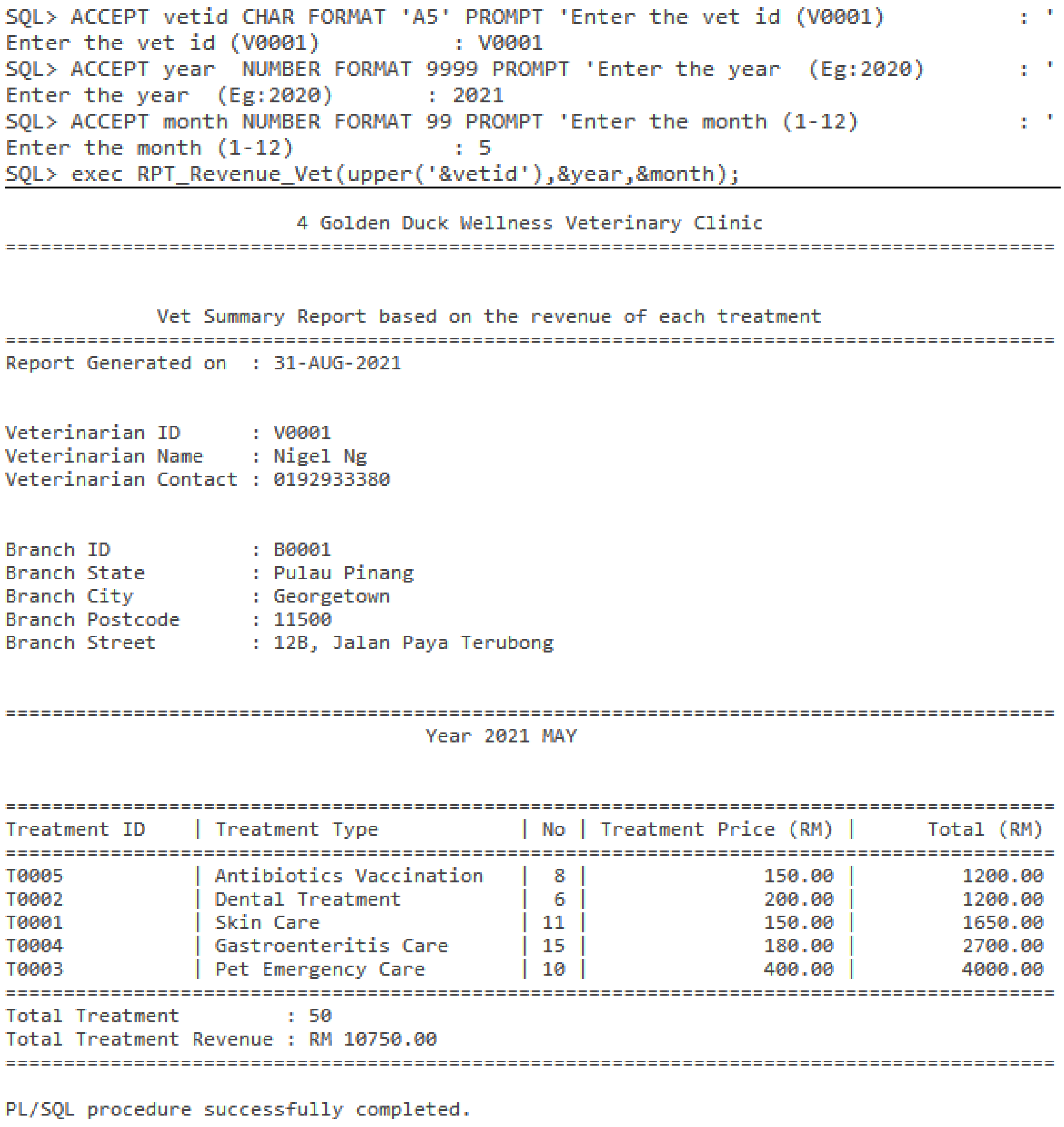
SQL statement:

```
CREATE OR REPLACE PROCEDURE RPT_Revenue_Vet(IN_vetid IN
VETERINARIAN.vet_id%TYPE ,IN_year IN NUMBER, IN_month IN
NUMBER) IS

 v_vetcount          NUMBER := 0;
 v_treatmentcount    NUMBER := 0;
 v_vetID             VETERINARIAN.vet_id%TYPE;
 v_vetname           VETERINARIAN.vet_name%TYPE;
 v_vetcontact        VETERINARIAN.vet_contact%TYPE;
 v_branchid          BRANCH.branch_id%TYPE;
 v_branchS           BRANCH.state%TYPE;
 v_branchC           BRANCH.city%TYPE;
 v_branchP           BRANCH.postcode%TYPE;
 v_branchST          BRANCH.streetname%TYPE;
 v_sumQuantity       NUMBER(10,2) := 0;
 v_sumRevenue        NUMBER(10,2) := 0;
 v_month             VARCHAR2(11);
 v_maxYear           NUMBER(4);
 v_minYear           NUMBER(4);
 v_maxMonth          NUMBER;
 v_sysdate           DATE;
 e_invalid_vetid EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_vetid, -20162);
 e_invalid_month EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_month, -20163);
 e_invalid_year EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_year, -20164);


CURSOR Treatment_CURSOR IS
SELECT T.treatment_id, T.treatment_type,
count(A.appointment_id) AS Number_of_Treatment,
t.treatment_price,
(count(A.appointment_id)*T.treatment_price) AS
TotalAmount
FROM Appointment A, Treatment T, Veterinarian V
WHERE A.treatment_id = T.treatment_id AND V.vet_id =
A.vet_id
      AND EXTRACT(YEAR FROM appointment_datetime) =
IN_year
      AND EXTRACT(MONTH FROM appointment_datetime) =
IN_month
```

```
       AND A.vet_id = IN_vetid
GROUP BY T.treatment_id, T.treatment_type,
T.treatment_price
ORDER BY TotalAmount ASC;


BEGIN

      DBMS_OUTPUT.PUT_LINE(chr(10));
      DBMS_OUTPUT.PUT_LINE(LPAD('4 Golden Duck Wellness
Veterinary Clinic',65, ' '));
      DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
      DBMS_OUTPUT.PUT_LINE(chr(10));
      DBMS_OUTPUT.PUT_LINE(LPAD('Vet Summary Report based
on the revenue of each treatment',70, ' '));
      DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));


      Select count(V.vet_id)  INTO v_vetcount from
veterinarian V Where V.vet_id = IN_vetid;

      IF v_vetcount = 0 THEN
          RAISE_APPLICATION_ERROR(-20162, 'Invalid Vet
ID', true);
       END IF;

     IF IN_MONTH < 1 OR IN_MONTH > 12 THEN
         RAISE_APPLICATION_ERROR(-20163, 'Invalid Month',
true);
      END IF;

      Select V.vet_id,
V.vet_name,V.vet_contact,B.branch_id, B.state, B.city,
B.Postcode,B.streetname,
             Extract(Year FROM
Max(A.appointment_datetime)),Extract(Year FROM
Min(A.appointment_datetime))
             INTO v_vetID, v_vetname, v_vetcontact,
v_branchid, v_branchS, v_branchC, v_branchP,v_branchST,
                v_maxyear, v_minyear
      From veterinarian V, branch B, Appointment A
      Where A.vet_id = V.vet_id AND V.branch_id =
B.branch_id AND V.vet_id = IN_vetid
      group by V.vet_id,
V.vet_name,V.vet_contact,B.branch_id, B.state, B.city,
B.Postcode,B.streetname;

      IF IN_YEAR < v_minyear OR IN_YEAR > v_maxyear THEN
         RAISE_APPLICATION_ERROR(-20164, 'Invalid Year',
true);
      END IF;
      Select sysdate into v_sysdate from dual;
      DBMS_OUTPUT.PUT_LINE('Report Generated on  : ' ||
v_sysdate);
      DBMS_OUTPUT.PUT_LINE(chr(10));
      DBMS_OUTPUT.PUT_LINE('Veterinarian ID      : ' ||
v_vetID);
      DBMS_OUTPUT.PUT_LINE('Veterinarian Name    : ' ||
v_vetName);
      DBMS_OUTPUT.PUT_LINE('Veterinarian Contact : ' ||
v_vetContact);

      DBMS_OUTPUT.PUT_LINE(chr(10));
      DBMS_OUTPUT.PUT_LINE('Branch ID            : ' ||
v_branchid);
      DBMS_OUTPUT.PUT_LINE('Branch State         : ' ||
v_branchS);
      DBMS_OUTPUT.PUT_LINE('Branch City          : ' ||
v_branchC);
      DBMS_OUTPUT.PUT_LINE('Branch Postcode      : ' ||
v_branchP);
      DBMS_OUTPUT.PUT_LINE('Branch Street        : ' ||
v_branchS);
      DBMS_OUTPUT.PUT_LINE(chr(10));
      DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));

      SELECT TO_CHAR(TO_DATE(IN_MONTH, 'MM'), 'MONTH')
INTO v_month FROM DUAL;

      DBMS_OUTPUT.PUT_LINE(LPAD('Year ' || IN_YEAR || '
'|| v_month,55,' '));
      DBMS_OUTPUT.PUT_LINE(chr(10));

      DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
      DBMS_OUTPUT.PUT_LINE(RPAD('Treatment ID',15,' ')||
' | ' || RPAD('Treatment Type',25,' ')|| ' | '
||LPAD('No',2,' ')|| ' | ' ||LPAD('Treatment Price
(RM)',20,' ')|| ' | ' ||LPAD('Total (RM)',15,' '));
      DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
      FOR treatment_record IN Treatment_CURSOR LOOP

DBMS_OUTPUT.PUT_LINE(RPAD(treatment_record.treatment_id,
15, ' ')|| ' | ' ||
RPAD(treatment_record.treatment_type,25,' ')||  ' | ' ||
LPAD(treatment_record.Number_of_Treatment,2,' ')
                             || ' | ' ||
LPAD(TO_CHAR(treatment_record.treatment_price,
999.99),20,' ')|| ' | ' ||
LPAD(to_char(treatment_record.TotalAmount,
'999999.99'),15,' '));
```

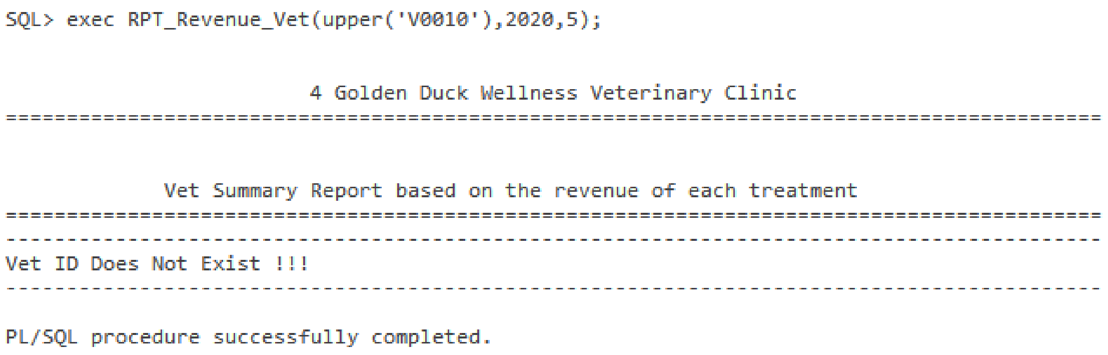
```
        v_sumQuantity := v_sumQuantity +
treatment_record.Number_of_Treatment;
        v_sumRevenue  := v_sumRevenue +
treatment_record.TotalAmount;
        v_treatmentcount := v_treatmentcount + 1;
    END LOOP;

    IF v_treatmentcount = 0 THEN
        DBMS_OUTPUT.PUT_LINE(chr(10));
        DBMS_OUTPUT.PUT_LINE(LPAD('No treatment handled
this month !!!',60,' '));
    END IF;

    DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));
    DBMS_OUTPUT.PUT_LINE('Total Treatment         : '||
v_sumQuantity);
    DBMS_OUTPUT.PUT_LINE('Total Treatment Revenue :
RM'|| to_char(v_sumRevenue,99999.99));
    DBMS_OUTPUT.PUT_LINE(LPAD('=',90,'='));

    EXCEPTION
     WHEN e_invalid_vetid THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Vet ID Does Not Exist
!!!');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
     WHEN e_invalid_month THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Invalid Month must be
within (1-12)');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
     WHEN e_invalid_year THEN
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));
      DBMS_OUTPUT.PUT_LINE ('Invalid Year or No
Treatment by '|| v_vetid || ' : ' || v_vetname || ' in
this year');
      DBMS_OUTPUT.PUT_LINE(LPAD('-',90,'-'));

END;
/

alter session set nls_date_format = 'DD-MON-YYYY';
ACCEPT vetid CHAR FORMAT 'A5' PROMPT 'Enter the vet id
(V0001)          : '
ACCEPT year  NUMBER FORMAT 9999 PROMPT 'Enter the year
(Eg:2020)        : '
ACCEPT month NUMBER FORMAT 99 PROMPT 'Enter the month
(1-12)           : '
exec RPT_Revenue_Vet(upper('&vetid'),&year,&month);
```

Sample Output:



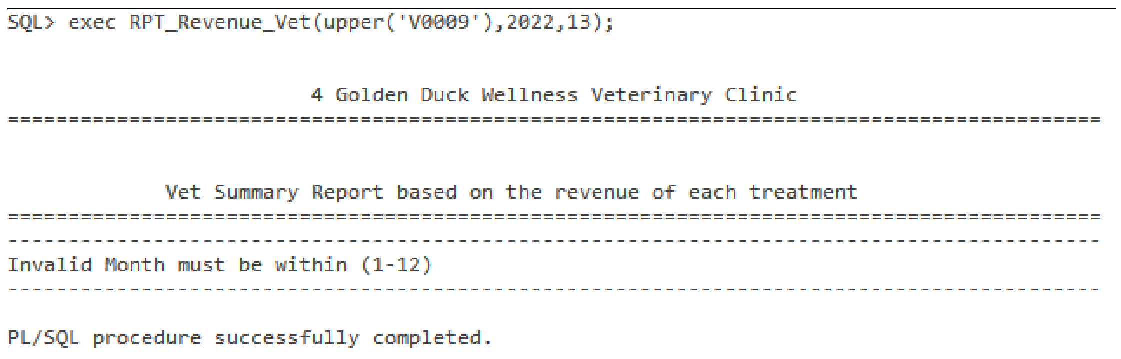
```
SQL> ACCEPT vetid CHAR FORMAT 'A5' PROMPT 'Enter the vet id (V0001)          : '
Enter the vet id (V0001)          : V0001
SQL> ACCEPT year  NUMBER FORMAT 9999 PROMPT 'Enter the year  (Eg:2020)       : '
Enter the year  (Eg:2020)       : 2021
SQL> ACCEPT month NUMBER FORMAT 99 PROMPT 'Enter the month (1-12)            : '
Enter the month (1-12)            : 5
SQL> exec RPT_Revenue_Vet(upper('&vetid'),&year,&month);

                         4 Golden Duck Wellness Veterinary Clinic
==========================================================================================


             Vet Summary Report based on the revenue of each treatment
==========================================================================================
Report Generated on  : 31-AUG-2021


Veterinarian ID      : V0001
Veterinarian Name    : Nigel Ng
Veterinarian Contact : 0192933380


Branch ID            : B0001
Branch State         : Pulau Pinang
Branch City          : Georgetown
Branch Postcode      : 11500
Branch Street        : 12B, Jalan Paya Terubong


==========================================================================================
                                    Year 2021 MAY


==========================================================================================
Treatment ID     | Treatment Type            | No | Treatment Price (RM) |      Total (RM)
==========================================================================================
T0005            | Antibiotics Vaccination   |  8 |               150.00 |         1200.00
T0002            | Dental Treatment          |  6 |               200.00 |         1200.00
T0001            | Skin Care                 | 11 |               150.00 |         1650.00
T0004            | Gastroenteritis Care      | 15 |               180.00 |         2700.00
T0003            | Pet Emergency Care        | 10 |               400.00 |         4000.00
==========================================================================================
Total Treatment         : 50
Total Treatment Revenue : RM 10750.00
==========================================================================================

PL/SQL procedure successfully completed.
```

Exception Output : Invalid veterinarian ID



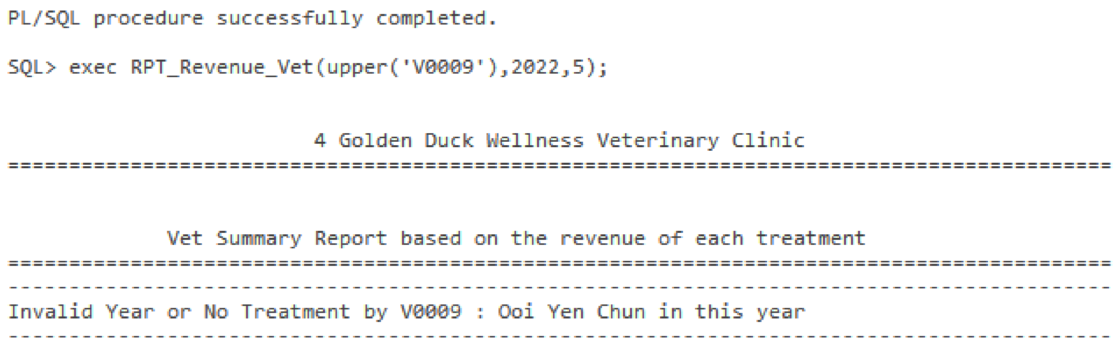
```
SQL> exec RPT_Revenue_Vet(upper('V0010'),2020,5);

                         4 Golden Duck Wellness Veterinary Clinic
==========================================================================================


             Vet Summary Report based on the revenue of each treatment
==========================================================================================
------------------------------------------------------------------------------------------
Vet ID Does Not Exist !!!
------------------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

Exception Output : Invalid Month



```
SQL> exec RPT_Revenue_Vet(upper('V0009'),2022,13);

                         4 Golden Duck Wellness Veterinary Clinic
==========================================================================================


             Vet Summary Report based on the revenue of each treatment
==========================================================================================
------------------------------------------------------------------------------------------
Invalid Month must be within (1-12)
------------------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

Exception Output : Invalid year or no treatment for the veterinarian in that month and year



```
PL/SQL procedure successfully completed.

SQL> exec RPT_Revenue_Vet(upper('V0009'),2022,5);

                         4 Golden Duck Wellness Veterinary Clinic
===========================================================================================


             Vet Summary Report based on the revenue of each treatment
===========================================================================================
-------------------------------------------------------------------------------------------
Invalid Year or No Treatment by V0009 : Ooi Yen Chun in this year
-------------------------------------------------------------------------------------------
```

4.4.9 Report 2: On demand report of days that performed poorly for a specific with a date range

Purpose: The purpose of this report is to analyze days that performed poorly within a date range. For example, Branch "B0001" from 1-MAY-2021 to 31-MAY-2021 which days that the brunch has revenue lower than RM 2000. Then a list of days with all the transactions will be displayed. At the end of the report will show which day has the revenue that is less than RM 2000. For example, within the date range two days that have lower revenue are on revenue. This allows the branch head to know which specific day has lower revenue and is required to improve it.

SQL statement:

```
CREATE OR REPLACE PROCEDURE RPT_Less_Revenue(IN_branchID
IN Transaction.branch_id%TYPE ,IN_StartDate in DATE,
IN_EndDate in DATE, IN_MinAmount in Number) IS

CURSOR less_transaction_CURSOR IS
select trunc(transaction_datetime) as
TransactionDate,to_char(transaction_datetime, 'DAY') AS
Day,
       count(transaction_id) AS TotalTransaction,
       sum(total_amount) AS Total_Amount,
((sum(total_amount))/(count(transaction_id))) AS
AverageT
from transaction
where transaction_datetime between IN_StartDate AND
IN_EndDate AND branch_id = IN_branchID
Having sum(total_amount) < IN_MinAmount
group by trunc(transaction_datetime),
to_char(transaction_datetime, 'DAY')
order by 1;

CURSOR transactiondetail_CURSOR (v_transaction_datetime
DATE) IS
select T.transaction_id , T.transaction_datetime,
TT.treatment_id, TT.treatment_type, TT.treatment_price,
       TD.medic_id, M.medic_name,
TD.line_qty,M.medic_price, TD.line_total
from transaction T, appointment A, treatment TT,
transactiondetail TD, medicalsupply M
where T.appointment_id = A.appointment_id AND
A.treatment_id = TT.treatment_id  AND
      TD.transaction_id = T.transaction_id AND
TD.medic_id = M.medic_id AND
      trunc(T.transaction_datetime) =
v_transaction_datetime AND T.branch_id = IN_branchID
order by 1;

CURSOR less_day_CURSOR IS
```

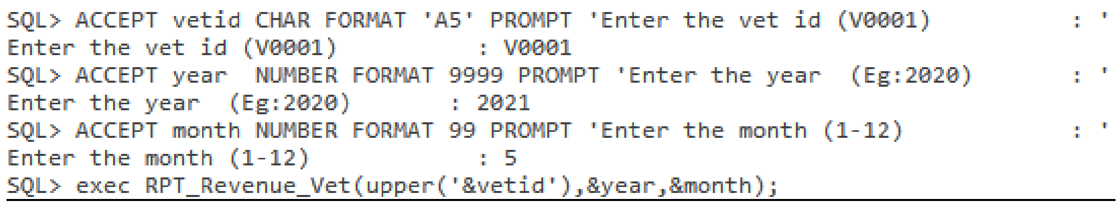
```
select to_char(DateTime, 'DAY') AS Day, count(*) AS
TotalDay
from (select trunc(transaction_datetime) AS DateTime,
sum(total_amount) as Total_amount
      from Transaction
      Where transaction_datetime Between '01-MAY-2021'
AND '31-MAY-2021' AND branch_id = 'B0001'
      group by trunc(transaction_datetime)
      having sum(total_amount) < 2000)
where Total_amount < 3000
group by to_char(DateTime, 'DAY')
order by 2;

 v_totalCount         NUMBER := 0;
 v_detailCount        NUMBER := 0;
 v_transactionid      TRANSACTION.transaction_id%type :=
'TPP00000000';
 v_rowCount           NUMBER := 0;
 v_sysdate            DATE;
 v_datecount          NUMBER :=0;
 v_branchcount        NUMBER :=0;
 e_invalid_branch EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_branch, -20165);


BEGIN
     DBMS_OUTPUT.PUT_LINE(chr(10));
     DBMS_OUTPUT.PUT_LINE(LPAD('4 Golden Duck Wellness
Veterinary Clinic',95, ' '));
     DBMS_OUTPUT.PUT_LINE(LPAD('=',150,'='));
     DBMS_OUTPUT.PUT_LINE(chr(10));
     DBMS_OUTPUT.PUT_LINE(LPAD('On Demand Report', 85, '
'));
     DBMS_OUTPUT.PUT_LINE(LPAD('Days that have less
revenue RM'||to_char(IN_MinAmount,999999.99),95, ' '));
     DBMS_OUTPUT.PUT_LINE(LPAD('FROM '|| IN_StartDate
||' to '||  IN_EndDate,90, ' '));
     Select sysdate into v_sysdate from dual;
     DBMS_OUTPUT.PUT_LINE('Report Generated on  : ' ||
v_sysdate);
     DBMS_OUTPUT.PUT_LINE(LPAD('=',150,'='));

     Select count(*) into v_branchcount from branch
where branch_id = IN_branchID;
     IF v_branchcount = 0 THEN
        RAISE_APPLICATION_ERROR(-20165, 'Invalid Branch
ID', true);
      END IF;

     FOR less_transaction_record IN
less_transaction_CURSOR LOOP
```

```
        DBMS_OUTPUT.PUT_LINE(RPAD('DATE : ' ||
less_transaction_record.TransactionDate,23, ' ')||
                             LPAD('| DAY : ' ||
less_transaction_record.Day,20, ' ') ||
                             LPAD('| No Transaction :'
||
to_char(less_transaction_record.TotalTransaction,99),25,
' ') ||
                             LPAD('| Total Amount RM '||
to_char(less_transaction_record.Total_Amount,9999.99),30
,' ')||
                             LPAD('| Average Amount RM
'||
to_char(less_transaction_record.AverageT,999.99),30,'
'));
        DBMS_OUTPUT.PUT_LINE(LPAD('=',150,'='));

        FOR transactiondetail_record IN
transactiondetail_CURSOR
(less_transaction_record.TransactionDate) LOOP
           IF v_transactionid !=
transactiondetail_record.transaction_id THEN
              DBMS_OUTPUT.PUT_LINE(chr(10));
              v_transactionid :=
transactiondetail_record.transaction_id;
              DBMS_OUTPUT.PUT_LINE(LPAD('-',150,'-'));
              DBMS_OUTPUT.PUT_LINE(RPAD('Transaction
ID',15, ' ')|| ' | ' ||RPAD('Treatment ID',12, ' ')|| '
| ' ||
                                   LPAD('Treatment
Type',25, ' ')|| ' | ' ||LPAD('Treatment Price',15, '
')|| ' | ' ||
                                   LPAD('Medicine
ID',12, ' ')|| ' | ' || LPAD('Medicine Name',17, ' ')||
' | ' ||
                                   LPAD('QTY',3, ' ')||
' | ' ||LPAD('Medicine Price',15, ' ')|| ' | '
||LPAD('Total (RM)',10, ' '));
              DBMS_OUTPUT.PUT_LINE(LPAD('-',150,'-'));


DBMS_OUTPUT.PUT_LINE(RPAD(transactiondetail_record.Trans
action_id,15, ' ')||' | ' ||
RPAD(transactiondetail_record.Treatment_id,12, ' ')||' |
' ||

LPAD(transactiondetail_record.treatment_type,25, ' ')||'
| '
||LPAD(to_char(transactiondetail_record.treatment_price,
999.99),15, ' ')||' | '||
LPAD(transactiondetail_record.medic_id,12, ' ')||' | '
|| LPAD(transactiondetail_record.medic_name,17, ' ')||'
| ' ||

LPAD(to_char(transactiondetail_record.line_qty,99),3, '
') ||' | ' ||
LPAD(to_char(transactiondetail_record.medic_price,
999.99),15, ' ') ||' | ' ||

LPAD(to_char(transactiondetail_record.line_total,9999.99
),10, ' '));
           ELSE

DBMS_OUTPUT.PUT_LINE(LPAD(transactiondetail_record.medic
_id,91, ' ')|| ' | ' ||
LPAD(transactiondetail_record.medic_name,17, ' ')|| ' |
' ||

LPAD(to_char(transactiondetail_record.line_qty,99),3, '
')||' | '||
LPAD(to_char(transactiondetail_record.medic_price,
999.99),15, ' ')|| ' | ' ||

LPAD(to_char(transactiondetail_record.line_total,9999.99
),10, ' '));
           END IF;
           v_rowCount := v_detailcount + 1;
        END LOOP;
        v_totalcount := v_totalcount +1;
        DBMS_OUTPUT.PUT_LINE(chr(10)||chr(10));
        DBMS_OUTPUT.PUT_LINE(LPAD('=',150,'='));
     END LOOP;

     IF v_rowCount = 0 THEN
            DBMS_OUTPUT.PUT_LINE(chr(10));
            DBMS_OUTPUT.PUT_LINE(LPAD('No sales than
less RM '|| to_char(IN_MinAmount,99999.99) ||'than
during this period',70, ' '));
            DBMS_OUTPUT.PUT_LINE(chr(10));
     ELSE
            DBMS_OUTPUT.PUT_LINE(chr(10));
            DBMS_OUTPUT.PUT_LINE(LPAD('=',50,'='));
            DBMS_OUTPUT.PUT_LINE(RPAD('Count of days
from MONDAY to SUNDAY',50,' ')|| '|');
            DBMS_OUTPUT.PUT_LINE(LPAD('=',50,'='));
            DBMS_OUTPUT.PUT_LINE(RPAD('Day',20,' ')||' |
' ||RPAD('Number',27,' ') || '|');
            DBMS_OUTPUT.PUT_LINE(LPAD('-',50,'-'));
            FOR dayCount_record in less_day_CURSOR LOOP
```

```
DBMS_OUTPUT.PUT_LINE(RPAD(dayCount_record.Day,20, ' ')||
' | ' ||RPAD(dayCount_record.TotalDay,27, ' ')|| '|');
                 v_dateCount := v_dateCount +
dayCount_record.TotalDay;
            END LOOP;
            DBMS_OUTPUT.PUT_LINE(LPAD('=',50,'='));
            DBMS_OUTPUT.PUT_LINE('Total ' || v_dateCount
|| ' days Revenue Less Than RM' ||
to_char(IN_MinAmount,99999.99));
            DBMS_OUTPUT.PUT_LINE(LPAD('=',50,'='));
     END IF;

     EXCEPTION
      WHEN e_invalid_branch THEN
       DBMS_OUTPUT.PUT_LINE(LPAD('-',150,'-'));
       DBMS_OUTPUT.PUT_LINE ('Branch ID Does Not Exist
!!!');
       DBMS_OUTPUT.PUT_LINE(LPAD('-',150,'-'));

END;
/
```

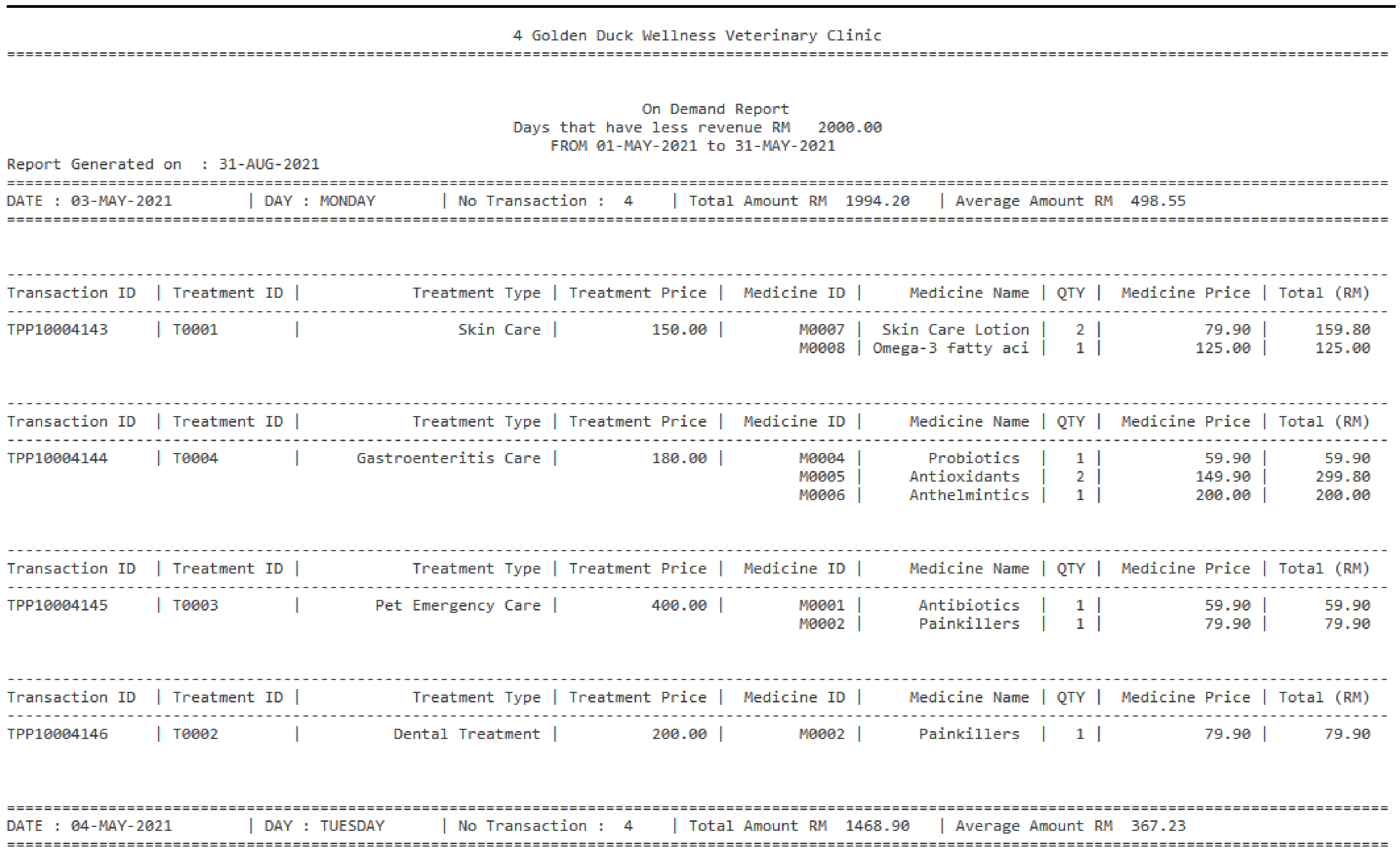
Exception Output: Out of clinic operated year.

Sample Output:

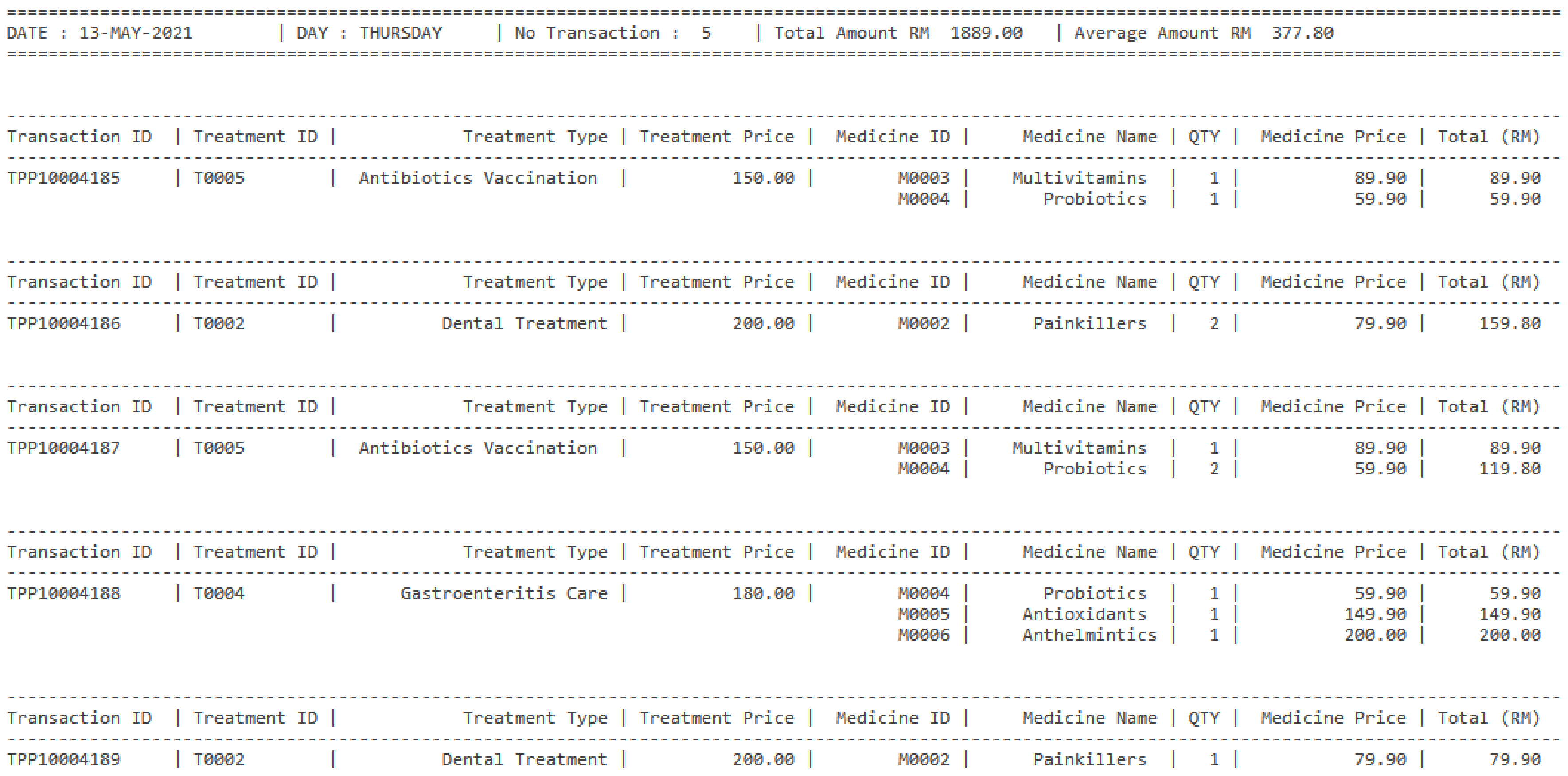


```
SQL> ACCEPT vetid CHAR FORMAT 'A5' PROMPT 'Enter the vet id (V0001)          : '
Enter the vet id (V0001)          : V0001
SQL> ACCEPT year  NUMBER FORMAT 9999 PROMPT 'Enter the year  (Eg:2020)       : '
Enter the year  (Eg:2020)       : 2021
SQL> ACCEPT month NUMBER FORMAT 99 PROMPT 'Enter the month (1-12)          : '
Enter the month (1-12)          : 5
SQL> exec RPT_Revenue_Vet(upper('&vetid'),&year,&month);
```

Sample Output :



```
                                                       4 Golden Duck Wellness Veterinary Clinic
======================================================================================================================================================


                                                                     On Demand Report
                                                       Days that have less revenue RM   2000.00
                                                           FROM 01-MAY-2021 to 31-MAY-2021
Report Generated on  : 31-AUG-2021
======================================================================================================================================================
DATE : 03-MAY-2021        | DAY : MONDAY       | No Transaction :  4    | Total Amount RM  1994.20   | Average Amount RM  498.55
======================================================================================================================================================


------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004143     | T0001        |                 Skin Care |          150.00 |        M0007 |  Skin Care Lotion |   2 |           79.90 |     159.80
                                                                                      M0008 | Omega-3 fatty aci |   1 |          125.00 |     125.00


------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004144     | T0004        |      Gastroenteritis Care |          180.00 |        M0004 |       Probiotics  |   1 |           59.90 |      59.90
                                                                                      M0005 |     Antioxidants  |   2 |          149.90 |     299.80
                                                                                      M0006 |     Anthelmintics |   1 |          200.00 |     200.00


------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004145     | T0003        |        Pet Emergency Care |          400.00 |        M0001 |      Antibiotics  |   1 |           59.90 |      59.90
                                                                                      M0002 |      Painkillers  |   1 |           79.90 |      79.90


------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004146     | T0002        |          Dental Treatment |          200.00 |        M0002 |      Painkillers  |   1 |           79.90 |      79.90


======================================================================================================================================================
DATE : 04-MAY-2021        | DAY : TUESDAY      | No Transaction :  4    | Total Amount RM  1468.90   | Average Amount RM  367.23
======================================================================================================================================================
=====================================================================================================================================
DATE : 04-MAY-2021          | DAY : TUESDAY       | No Transaction :  4     | Total Amount RM  1468.90    | Average Amount RM  367.23
=====================================================================================================================================


-------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |      Medicine Name | QTY |  Medicine Price | Total (RM)
-------------------------------------------------------------------------------------------------------------------------------------
TPP10004147     | T0005        |  Antibiotics Vaccination  |          150.00 |        M0003 |     Multivitamins  |   1 |           89.90 |      89.90
                                                                                    M0004 |        Probiotics  |   1 |           59.90 |      59.90


-------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |      Medicine Name | QTY |  Medicine Price | Total (RM)
-------------------------------------------------------------------------------------------------------------------------------------
TPP10004148     | T0005        |  Antibiotics Vaccination  |          150.00 |        M0003 |     Multivitamins  |   2 |           89.90 |     179.80
                                                                                    M0004 |        Probiotics  |   1 |           59.90 |      59.90


-------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |      Medicine Name | QTY |  Medicine Price | Total (RM)
-------------------------------------------------------------------------------------------------------------------------------------
TPP10004149     | T0005        |  Antibiotics Vaccination  |          150.00 |        M0003 |     Multivitamins  |   2 |           89.90 |     179.80
                                                                                    M0004 |        Probiotics  |   1 |           59.90 |      59.90


-------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |      Medicine Name | QTY |  Medicine Price | Total (RM)
-------------------------------------------------------------------------------------------------------------------------------------
TPP10004150     | T0005        |  Antibiotics Vaccination  |          150.00 |        M0003 |     Multivitamins  |   2 |           89.90 |     179.80
                                                                                    M0004 |        Probiotics  |   1 |           59.90 |      59.90
```

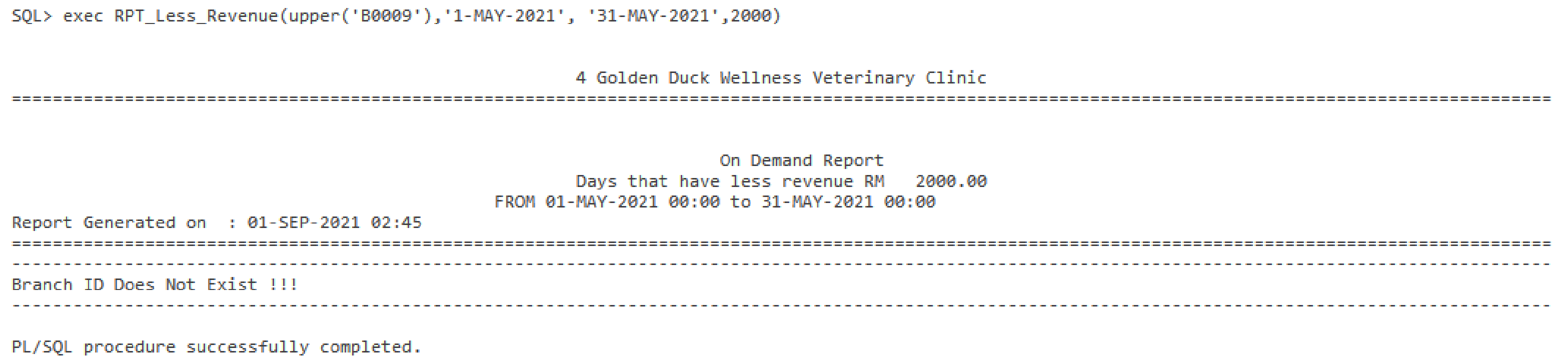


```
=========================================================================================================================================================
DATE : 13-MAY-2021          | DAY : THURSDAY      | No Transaction :  5     | Total Amount RM  1889.00    | Average Amount RM  377.80
=========================================================================================================================================================


---------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |             Treatment Type | Treatment Price |   Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
---------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004185     | T0005        |  Antibiotics Vaccination   |          150.00 |         M0003 |   Multivitamins   |   1 |           89.90 |      89.90
                                                                                       M0004 |      Probiotics   |   1 |           59.90 |      59.90


---------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |             Treatment Type | Treatment Price |   Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
---------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004186     | T0002        |          Dental Treatment  |          200.00 |         M0002 |     Painkillers   |   2 |           79.90 |     159.80


---------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |             Treatment Type | Treatment Price |   Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
---------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004187     | T0005        |  Antibiotics Vaccination   |          150.00 |         M0003 |   Multivitamins   |   1 |           89.90 |      89.90
                                                                                       M0004 |      Probiotics   |   2 |           59.90 |     119.80


---------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |             Treatment Type | Treatment Price |   Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
---------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004188     | T0004        |      Gastroenteritis Care  |          180.00 |         M0004 |      Probiotics   |   1 |           59.90 |      59.90
                                                                                       M0005 |    Antioxidants   |   1 |          149.90 |     149.90
                                                                                       M0006 |    Anthelmintics  |   1 |          200.00 |     200.00


---------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |             Treatment Type | Treatment Price |   Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
---------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004189     | T0002        |          Dental Treatment  |          200.00 |         M0002 |     Painkillers   |   1 |           79.90 |      79.90
```

```
=================================================================================================================================================================
DATE : 20-MAY-2021         | DAY : THURSDAY      | No Transaction :  4     | Total Amount RM  1989.30    | Average Amount RM  497.33
=================================================================================================================================================================


-----------------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |      Medicine Name | QTY |  Medicine Price | Total (RM)
-----------------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004221     | T0002        |          Dental Treatment |          200.00 |        M0002 |       Painkillers  |   2 |           79.90 |     159.80


-----------------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |      Medicine Name | QTY |  Medicine Price | Total (RM)
-----------------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004222     | T0004        |      Gastroenteritis Care |          180.00 |        M0004 |        Probiotics  |   1 |           59.90 |      59.90
                                                                                     M0005 |      Antioxidants  |   1 |          149.90 |     149.90
                                                                                     M0006 |      Anthelmintics |   2 |          200.00 |     400.00


-----------------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |      Medicine Name | QTY |  Medicine Price | Total (RM)
-----------------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004223     | T0001        |                 Skin Care |          150.00 |        M0007 |  Skin Care Lotion |   1 |           79.90 |      79.90
                                                                                     M0008 | Omega-3 fatty aci |   2 |          125.00 |     250.00


-----------------------------------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |      Medicine Name | QTY |  Medicine Price | Total (RM)
-----------------------------------------------------------------------------------------------------------------------------------------------------------------
TPP10004224     | T0002        |          Dental Treatment |          200.00 |        M0002 |       Painkillers  |   2 |           79.90 |     159.80
```

```
====================================================================================================================================
DATE : 22-MAY-2021        | DAY : SATURDAY      | No Transaction :  3     | Total Amount RM  1119.60    | Average Amount RM  373.20
====================================================================================================================================


------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
------------------------------------------------------------------------------------------------------------------------------------
TPP10004230     | T0002        |          Dental Treatment |          200.00 |        M0002 |       Painkillers  |  1 |           79.90 |      79.90


------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
------------------------------------------------------------------------------------------------------------------------------------
TPP10004231     | T0001        |                 Skin Care |          150.00 |        M0007 |  Skin Care Lotion |  1 |           79.90 |      79.90
                                                                                     M0008 | Omega-3 fatty aci |  2 |          125.00 |     250.00


------------------------------------------------------------------------------------------------------------------------------------
Transaction ID  | Treatment ID |            Treatment Type | Treatment Price |  Medicine ID |     Medicine Name | QTY |  Medicine Price | Total (RM)
------------------------------------------------------------------------------------------------------------------------------------
TPP10004232     | T0002        |          Dental Treatment |          200.00 |        M0002 |       Painkillers  |  2 |           79.90 |     159.80


====================================================================================================================================


==================================================
Count of days from MONDAY to SUNDAY               |
==================================================
Day                     | Number                  |
--------------------------------------------------
TUESDAY                 | 1                       |
MONDAY                  | 1                       |
SATURDAY                | 1                       |
THURSDAY                | 2                       |
==================================================
Total 5 days Revenue Less Than RM  2000.00
==================================================
```

Exception Output: Invalid Branch ID



```
SQL> exec RPT_Less_Revenue(upper('B0009'),'1-MAY-2021', '31-MAY-2021',2000)

                                                       4 Golden Duck Wellness Veterinary Clinic
======================================================================================================================================================


                                                                     On Demand Report
                                                       Days that have less revenue RM   2000.00
                                               FROM 01-MAY-2021 00:00 to 31-MAY-2021 00:00
Report Generated on  : 01-SEP-2021 02:45
======================================================================================================================================================
------------------------------------------------------------------------------------------------------------------------------------------------------
Branch ID Does Not Exist !!!
------------------------------------------------------------------------------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

4.4.10 Report 3: Detail report of each Branch's Performance in a specific year

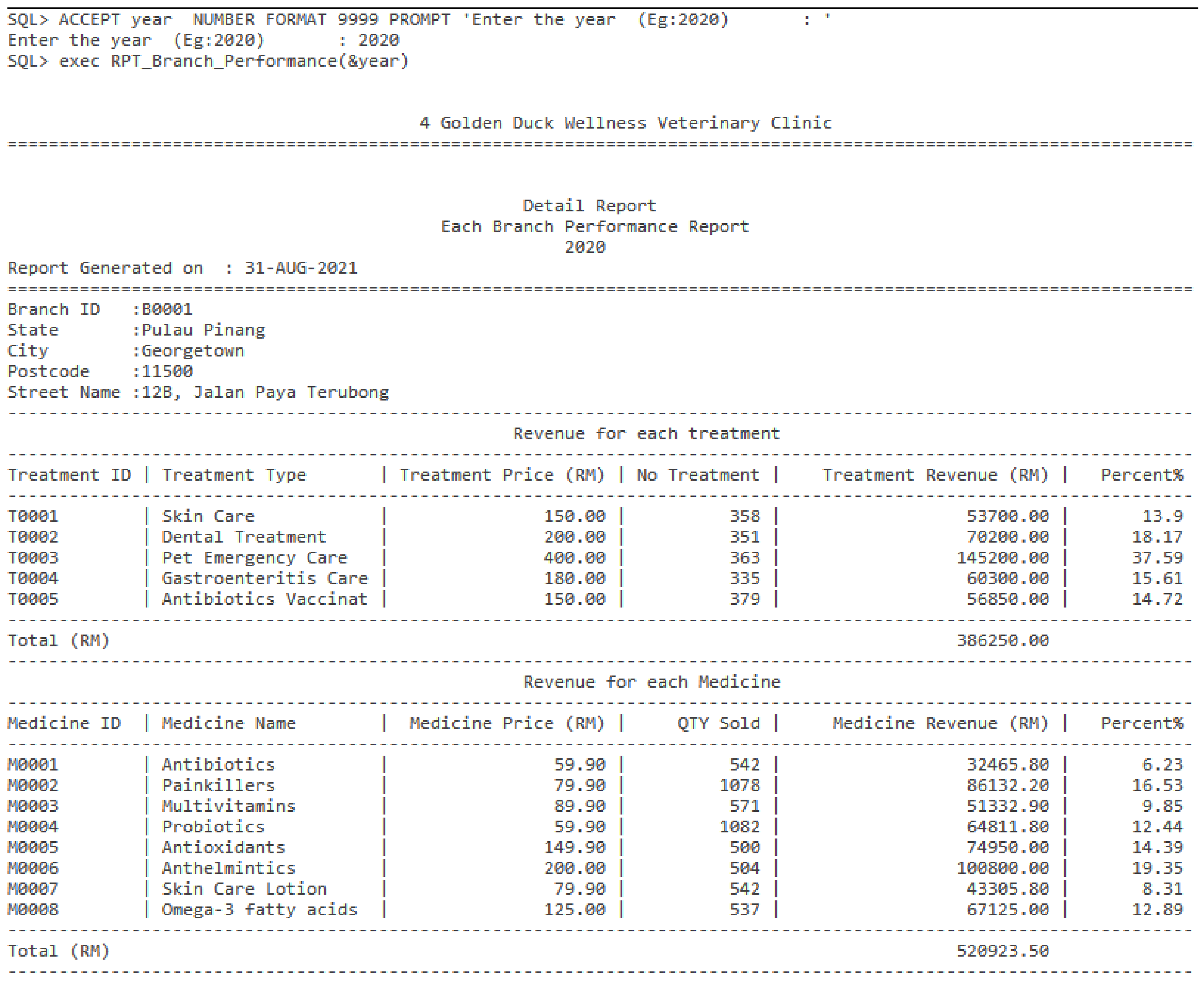
Purpose: The purpose of this report is to show total revenue, revenue of each treatment and revenue of each medecine of a branch. Details such as treatment done, medicine sold by each branch will also be shown, contribution of each treatment and medicine to the branch revenue will also be shown . The contribution of each branch to the total revenue of the specific year will be shown in percentage.

SQL statement:

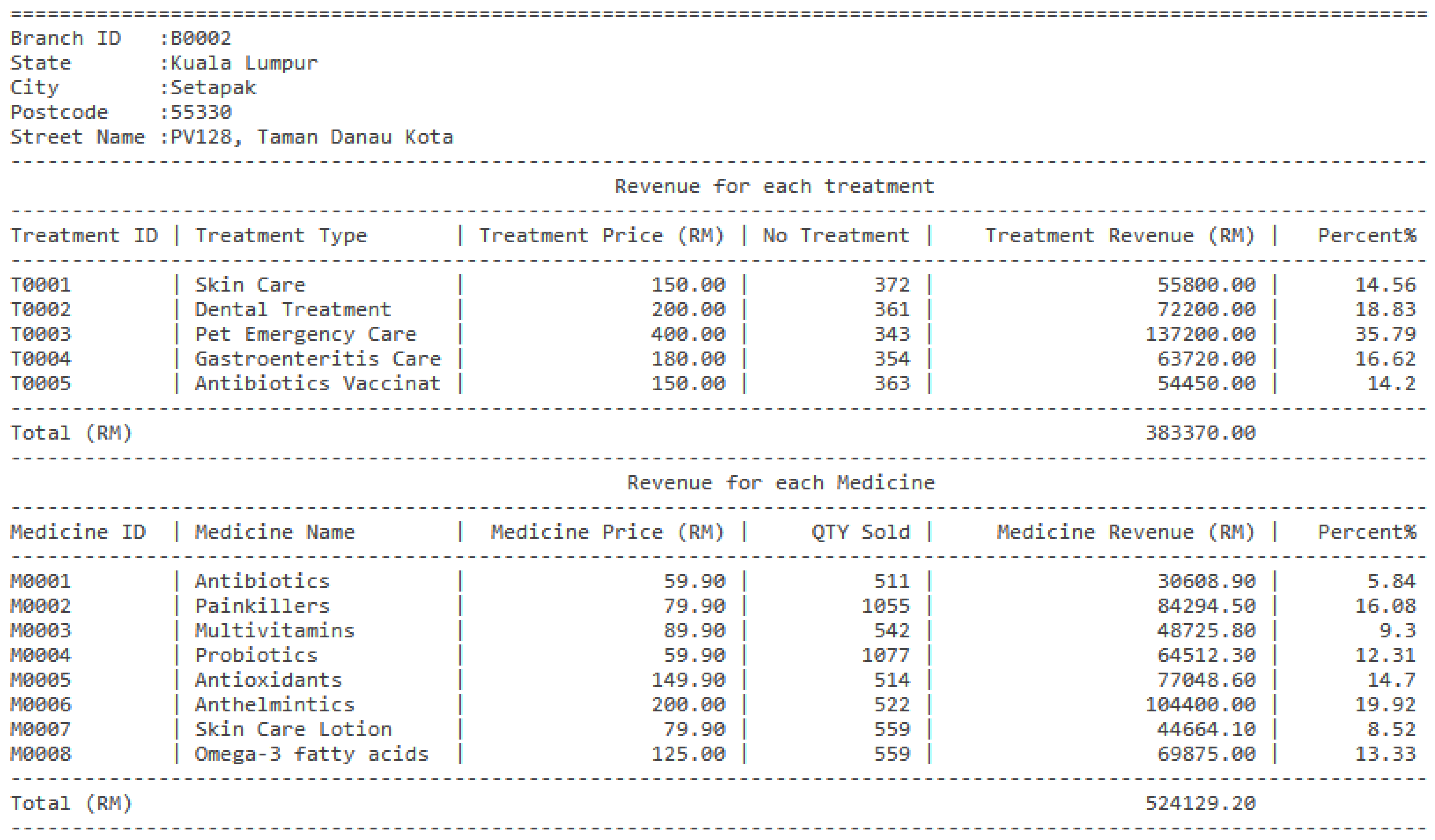
```
CREATE OR REPLACE PROCEDURE
RPT_Branch_Performance(in_YEAR IN NUMBER) IS

CURSOR branch_overall_revenue_CURSOR IS
select B.branch_id , B.state, B.city, B.postcode,
B.streetname,
       count(T.transaction_id) AS TotalTransaction,
       sum(T.total_amount) AS Total_Amount
from transaction T, Branch B
where T.branch_id = B.branch_id AND Extract(YEAR from
transaction_datetime) = 2020
group by B.branch_id , B.state, B.city, B.postcode,
B.streetname
order by 1;

CURSOR branch_treatment_revenue_CURSOR (v_branchID
BRANCH.branch_id%TYPE) IS
select B.branch_id , TT.treatment_id, TT.treatment_type,
TT.treatment_price,
       count(T.transaction_id) as TotalTreatment,
       (count(T.transaction_id)*treatment_price) AS
TotalTreatmentRevenue
from transaction T, Branch B, appointment A, treatment
TT
where T.branch_id = B.branch_id AND T.appointment_id =
A.appointment_id AND
      A.treatment_id = TT.treatment_id AND
      Extract(YEAR from transaction_datetime) = in_YEAR
AND
      T.branch_id = v_branchID
group by B.branch_id, TT.treatment_id,
TT.treatment_type, TT.treatment_price
order by 1,2;

CURSOR branch_medic_revenue_CURSOR (v_branchID
BRANCH.branch_id%TYPE) IS
select T.branch_id , M.medic_id, M.medic_name,
M.medic_price,
       sum(TD.line_qty) as TotalMedic,
       sum(TD.line_total) AS TotalMedicRevenue
from transaction T, transactiondetail TD, medicalsupply
M
where T.transaction_id = TD.transaction_id AND
TD.medic_id = M.medic_id AND
      Extract(YEAR from T.transaction_datetime) =
in_YEAR AND
      T.branch_id = v_branchID
group by T.branch_id, M.medic_id, M.medic_name,
M.medic_price
order by 1,2;

 v_branchTotalTreatmentRevenue         NUMBER := 0;
 v_branchTotalMedicRevenue             NUMBER := 0;
 v_all_totalamount                     NUMBER;
 v_all_totalTreatment                  NUMBER :=0;
 v_all_totalMedical                    NUMBER :=0;
 v_transactionid
TRANSACTION.transaction_id%type := 'TPP00000000';
 v_rowCount                            NUMBER := 0;
 v_sysdate                             DATE;
 v_branchcount                         NUMBER :=0;
 v_allRevenue                          NUMBER :=0;
 v_minYear                             NUMBER;
 v_maxYear                             NUMBER;
 e_invalid_year EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_year, -20166);


BEGIN
     DBMS_OUTPUT.PUT_LINE(chr(10));
     DBMS_OUTPUT.PUT_LINE(LPAD('4 Golden Duck Wellness
Veterinary Clinic',80, ' '));
     DBMS_OUTPUT.PUT_LINE(LPAD('=',115,'='));
     DBMS_OUTPUT.PUT_LINE(chr(10));
     DBMS_OUTPUT.PUT_LINE(LPAD('Detail Report', 63, '
'));
     DBMS_OUTPUT.PUT_LINE(LPAD('Each Branch Performance
Report',72, ' '));
     DBMS_OUTPUT.PUT_LINE(LPAD(IN_YEAR,58, ' '));
     Select sysdate into v_sysdate from dual;
     DBMS_OUTPUT.PUT_LINE('Report Generated on  : ' ||
v_sysdate);

     Select Extract(YEAR FROM
Min(transaction_datetime)), Extract(YEAR FROM
Max(transaction_datetime))
            into v_minYear, v_maxYear
     From transaction;

     IF in_YEAR < v_minYear or in_Year > v_maxYear THEN
```

```
          RAISE_APPLICATION_ERROR(-20166, 'Invalid
Year', true);
      END IF;

     FOR branch_record IN branch_overall_revenue_CURSOR
LOOP
        DBMS_OUTPUT.PUT_LINE(LPAD('=',115,'='));
        DBMS_OUTPUT.PUT_LINE('Branch ID   :' ||
branch_record.branch_id);
        DBMS_OUTPUT.PUT_LINE('State       :' ||
branch_record.state);
        DBMS_OUTPUT.PUT_LINE('City        :' ||
branch_record.city);
        DBMS_OUTPUT.PUT_LINE('Postcode    :' ||
branch_record.postcode);
        DBMS_OUTPUT.PUT_LINE('Street Name :' ||
branch_record.streetname);
        DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
        DBMS_OUTPUT.PUT_LINE(LPAD('Revenue for each
treatment',75,' '));
        DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
        DBMS_OUTPUT.PUT_LINE(RPAD('Treatment ID', 12 ,'
')|| ' | ' || RPAD('Treatment Type', 20 ,' ')|| ' | ' ||
                             LPAD('Treatment Price
(RM)', 20 ,' ')|| ' | ' ||
                             LPAD('No Treatment', 12 ,'
')|| ' | ' ||
                             LPAD('Treatment Revenue
(RM)', 25 ,' ')|| ' | ' ||
                             LPAD('Percent%',10,' '));
        DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
        FOR treatment_record1 IN
branch_treatment_revenue_CURSOR
(branch_record.branch_id) LOOP
            v_branchTotalTreatmentRevenue :=
v_branchTotalTreatmentRevenue +
treatment_record1.TotalTreatmentRevenue;
        END LOOP;
        FOR treatment_record IN
branch_treatment_revenue_CURSOR
(branch_record.branch_id) LOOP

DBMS_OUTPUT.PUT_LINE(RPAD(treatment_record.treatment_id,
12 ,' ')|| ' | ' ||

RPAD(treatment_record.treatment_type, 20 ,' ')|| ' | '
||

LPAD(to_char(treatment_record.treatment_price, 999.99),
20 ,' ')|| ' | ' ||
LPAD(treatment_record.TotalTreatment, 12 ,' ')|| ' | '
||

LPAD(to_char(treatment_record.TotalTreatmentRevenue,
999999.99), 25 ,' ')|| ' | ' ||

LPAD(round((treatment_record.TotalTreatmentRevenue/v_bra
nchTotalTreatmentRevenue*100),2),10,' '));
        END LOOP;
        DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
        DBMS_OUTPUT.PUT_LINE('Total
(RM)'||LPAD(to_char(v_branchTotalTreatmentRevenue,999999
99.99),91,' '));
        DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
        DBMS_OUTPUT.PUT_LINE(LPAD('Revenue for each
Medicine',75,' '));
        DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
        DBMS_OUTPUT.PUT_LINE(RPAD('Medicine ID', 12 ,'
')|| ' | ' || RPAD('Medicine Name', 20 ,' ')|| ' | ' ||
                             LPAD('Medicine Price (RM)',
20 ,' ')|| ' | ' ||
                             LPAD('QTY Sold', 12 ,' ')||
' | ' ||
                             LPAD('Medicine Revenue
(RM)', 25 ,' ')|| ' | ' ||
                             LPAD('Percent%',10,' '));
        DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
         FOR medic_record1 IN
branch_medic_revenue_CURSOR (branch_record.branch_id)
LOOP
            v_branchTotalMedicRevenue :=
v_branchTotalMedicRevenue +
medic_record1.TotalMedicRevenue;
         END LOOP;
         FOR medic_record IN branch_medic_revenue_CURSOR
(branch_record.branch_id) LOOP

DBMS_OUTPUT.PUT_LINE(RPAD(medic_record.medic_id, 12 ,'
')|| ' | ' ||

RPAD(medic_record.medic_name, 20 ,' ')|| ' | ' ||

LPAD(to_char(medic_record.medic_price, 999.99), 20 ,'
')|| ' | ' ||

LPAD(medic_record.TotalMedic, 12 ,' ')|| ' | ' ||

LPAD(to_char(medic_record.TotalMedicRevenue, 999999.99),
25 ,' ')|| ' | ' ||
LPAD(round((medic_record.TotalMedicRevenue/v_branchTotal
MedicRevenue*100),2),10,' '));
         END LOOP;
          DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
          DBMS_OUTPUT.PUT_LINE('Total
(RM)'||LPAD(to_char(v_branchTotalMedicRevenue,99999999.9
9),91,' '));

         v_branchTotalTreatmentRevenue:= 0;
         v_branchTotalMedicRevenue:= 0;
         DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
         DBMS_OUTPUT.PUT_LINE(chr(10));

     END LOOP;

     DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
     DBMS_OUTPUT.PUT_LINE(LPAD('Contribution of each
branch in ',70,' ' )|| IN_YEAR);
     DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
     DBMS_OUTPUT.PUT_LINE(RPAD('Branch ID ', 10, ' ')
||' | ' || LPAD('Treatment Revenue (RM)',25,' ') || ' |
' ||
                         LPAD('Medicine Revenue
(RM)',25,' ') ||' | ' || LPAD('Branch Total Revenue
(RM)',25,' ') ||' | ' || LPAD('Contribution%',17,' '));
     DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));

     Select sum(total_amount) into v_all_totalamount
from transaction where Extract(YEAR from
transaction_datetime) = IN_year;
     FOR branch_record IN branch_overall_revenue_CURSOR
LOOP
       v_branchTotalTreatmentRevenue:= 0;
       v_branchTotalMedicRevenue:= 0;
       FOR treatment_record1 IN
branch_treatment_revenue_CURSOR
(branch_record.branch_id) LOOP
            v_branchTotalTreatmentRevenue :=
v_branchTotalTreatmentRevenue +
treatment_record1.TotalTreatmentRevenue;
         END LOOP;
       FOR medic_record1 IN branch_medic_revenue_CURSOR
(branch_record.branch_id) LOOP
            v_branchTotalMedicRevenue :=
v_branchTotalMedicRevenue +
medic_record1.TotalMedicRevenue;
        END LOOP;

DBMS_OUTPUT.PUT_LINE(RPAD(branch_record.branch_id, 10, '
') ||' | ' ||
LPAD(to_char(v_branchTotalTreatmentRevenue,999999.99),25
,' ') || ' | ' ||

LPAD(to_char(v_branchTotalMedicRevenue, 999999.99),25, '
') ||' | ' ||

LPAD(to_char(branch_record.Total_Amount, 999999.99),25,'
') ||' | ' ||

LPAD(round((branch_record.Total_Amount/v_all_totalamount
*100),2),17,' '));
       v_all_totalTreatment := v_all_totalTreatment +
v_branchTotalTreatmentRevenue;
       v_all_totalMedical   := v_all_totalMedical +
v_branchTotalMedicRevenue;
     END LOOP;
       DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
       DBMS_OUTPUT.PUT_LINE('Total (RM) |'||
LPAD(to_char(v_all_totalTreatment,9999999.99),26,' ')
||' | ' ||

LPAD(to_char(v_all_totalMedical,9999999.99),25,' ')||' |
' ||
                            LPAD(to_char(
v_all_totalamount,9999999.99),25,' ')||' | ' );
       DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
     EXCEPTION
      WHEN e_invalid_year THEN
       DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));
       DBMS_OUTPUT.PUT_LINE ('The year you have entered
is invalid or within the range of operated year !!!');
       DBMS_OUTPUT.PUT_LINE(LPAD('-',115,'-'));

END ;
/

exec RPT_Branch_Performance(&year)
```

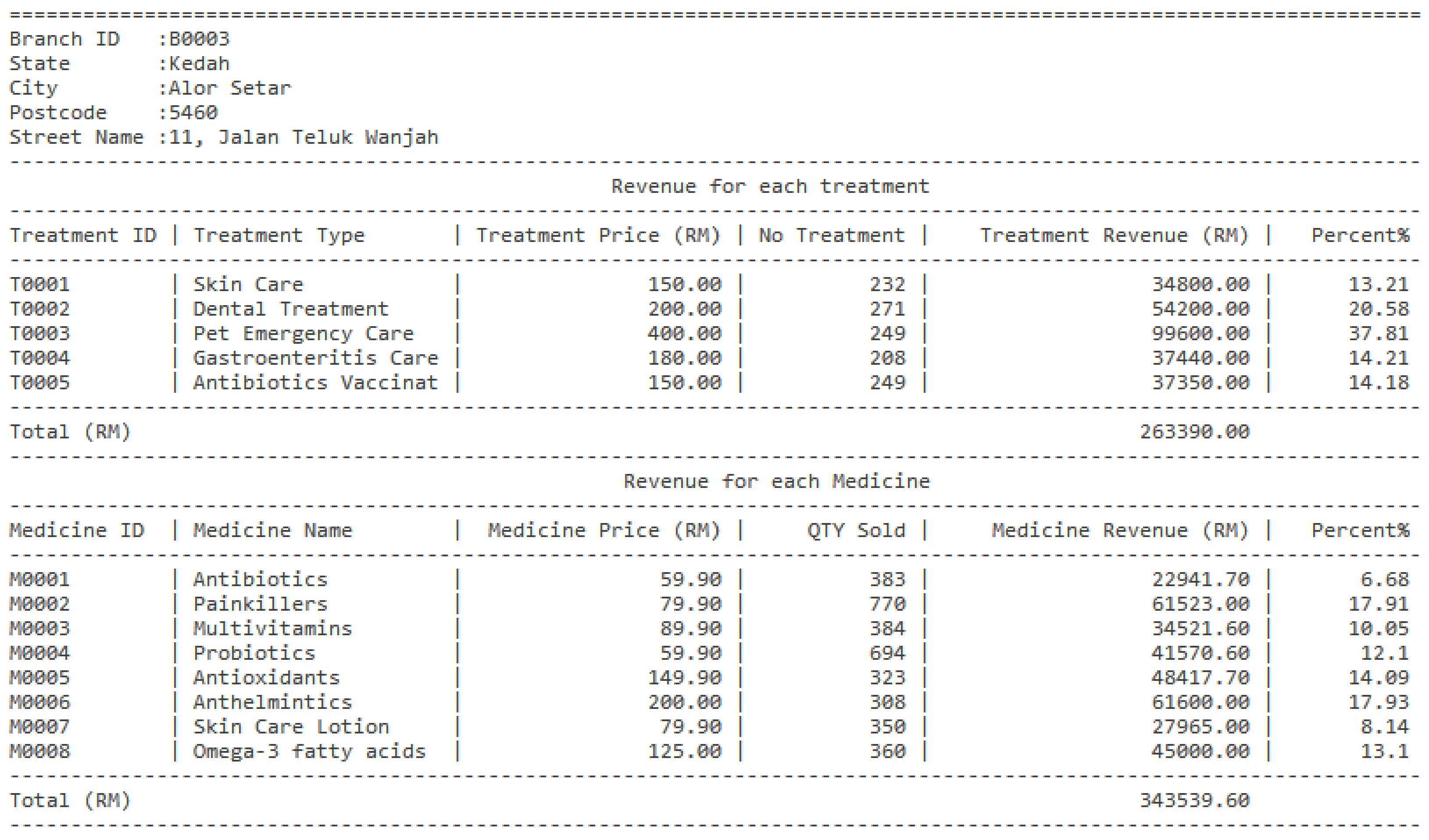
Sample Output :



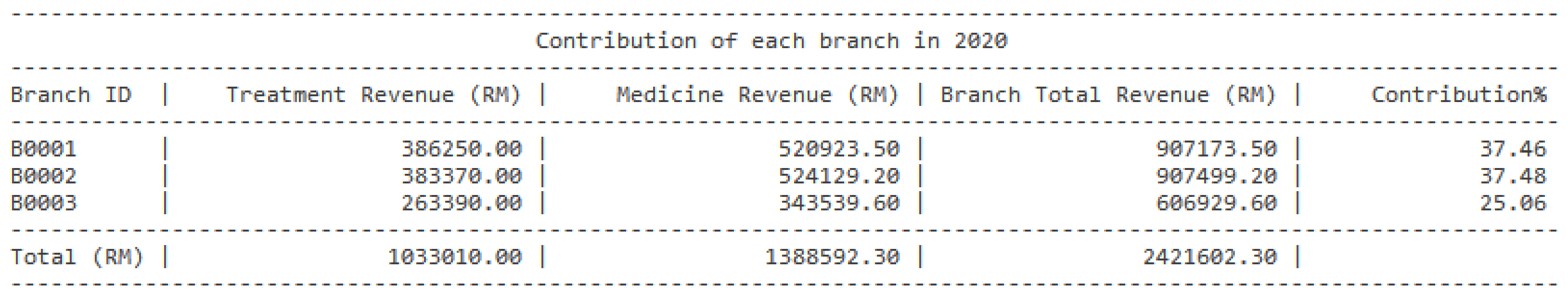
```
SQL> ACCEPT year  NUMBER FORMAT 9999 PROMPT 'Enter the year  (Eg:2020)       : '
Enter the year  (Eg:2020)       : 2020
SQL> exec RPT_Branch_Performance(&year)


                                        4 Golden Duck Wellness Veterinary Clinic
==================================================================================================================


                                                  Detail Report
                                          Each Branch Performance Report
                                                      2020
Report Generated on  : 31-AUG-2021
==================================================================================================================
Branch ID   :B0001
State       :Pulau Pinang
City        :Georgetown
Postcode    :11500
Street Name :12B, Jalan Paya Terubong
------------------------------------------------------------------------------------------------------------------
                                                 Revenue for each treatment
------------------------------------------------------------------------------------------------------------------
Treatment ID | Treatment Type        | Treatment Price (RM) | No Treatment |    Treatment Revenue (RM) |    Percent%
------------------------------------------------------------------------------------------------------------------
T0001        | Skin Care             |               150.00 |          358 |                  53700.00 |        13.9
T0002        | Dental Treatment      |               200.00 |          351 |                  70200.00 |       18.17
T0003        | Pet Emergency Care    |               400.00 |          363 |                 145200.00 |       37.59
T0004        | Gastroenteritis Care  |               180.00 |          335 |                  60300.00 |       15.61
T0005        | Antibiotics Vaccinat  |               150.00 |          379 |                  56850.00 |       14.72
------------------------------------------------------------------------------------------------------------------
Total (RM)                                                                               386250.00
------------------------------------------------------------------------------------------------------------------
                                                  Revenue for each Medicine
------------------------------------------------------------------------------------------------------------------
Medicine ID  | Medicine Name         |  Medicine Price (RM) |     QTY Sold |     Medicine Revenue (RM) |    Percent%
------------------------------------------------------------------------------------------------------------------
M0001        | Antibiotics           |                59.90 |          542 |                  32465.80 |        6.23
M0002        | Painkillers           |                79.90 |         1078 |                  86132.20 |       16.53
M0003        | Multivitamins         |                89.90 |          571 |                  51332.90 |        9.85
M0004        | Probiotics            |                59.90 |         1082 |                  64811.80 |       12.44
M0005        | Antioxidants          |               149.90 |          500 |                  74950.00 |       14.39
M0006        | Anthelmintics         |               200.00 |          504 |                 100800.00 |       19.35
M0007        | Skin Care Lotion      |                79.90 |          542 |                  43305.80 |        8.31
M0008        | Omega-3 fatty acids   |               125.00 |          537 |                  67125.00 |       12.89
------------------------------------------------------------------------------------------------------------------
Total (RM)                                                                               520923.50
------------------------------------------------------------------------------------------------------------------
```



```
===================================================================================================================
Branch ID   :B0002
State       :Kuala Lumpur
City        :Setapak
Postcode    :55330
Street Name :PV128, Taman Danau Kota
-------------------------------------------------------------------------------------------------------------------
                                                 Revenue for each treatment
-------------------------------------------------------------------------------------------------------------------
Treatment ID | Treatment Type        | Treatment Price (RM) | No Treatment |    Treatment Revenue (RM) |    Percent%
-------------------------------------------------------------------------------------------------------------------
T0001        | Skin Care             |               150.00 |          372 |                  55800.00 |       14.56
T0002        | Dental Treatment      |               200.00 |          361 |                  72200.00 |       18.83
T0003        | Pet Emergency Care    |               400.00 |          343 |                 137200.00 |       35.79
T0004        | Gastroenteritis Care  |               180.00 |          354 |                  63720.00 |       16.62
T0005        | Antibiotics Vaccinat  |               150.00 |          363 |                  54450.00 |        14.2
-------------------------------------------------------------------------------------------------------------------
Total (RM)                                                                              383370.00
-------------------------------------------------------------------------------------------------------------------
                                                  Revenue for each Medicine
-------------------------------------------------------------------------------------------------------------------
Medicine ID  | Medicine Name         |  Medicine Price (RM) |     QTY Sold |     Medicine Revenue (RM) |    Percent%
-------------------------------------------------------------------------------------------------------------------
M0001        | Antibiotics           |                59.90 |          511 |                  30608.90 |        5.84
M0002        | Painkillers           |                79.90 |         1055 |                  84294.50 |       16.08
M0003        | Multivitamins         |                89.90 |          542 |                  48725.80 |         9.3
M0004        | Probiotics            |                59.90 |         1077 |                  64512.30 |       12.31
M0005        | Antioxidants          |               149.90 |          514 |                  77048.60 |        14.7
M0006        | Anthelmintics         |               200.00 |          522 |                 104400.00 |       19.92
M0007        | Skin Care Lotion      |                79.90 |          559 |                  44664.10 |        8.52
M0008        | Omega-3 fatty acids   |               125.00 |          559 |                  69875.00 |       13.33
-------------------------------------------------------------------------------------------------------------------
Total (RM)                                                                              524129.20
-------------------------------------------------------------------------------------------------------------------
```

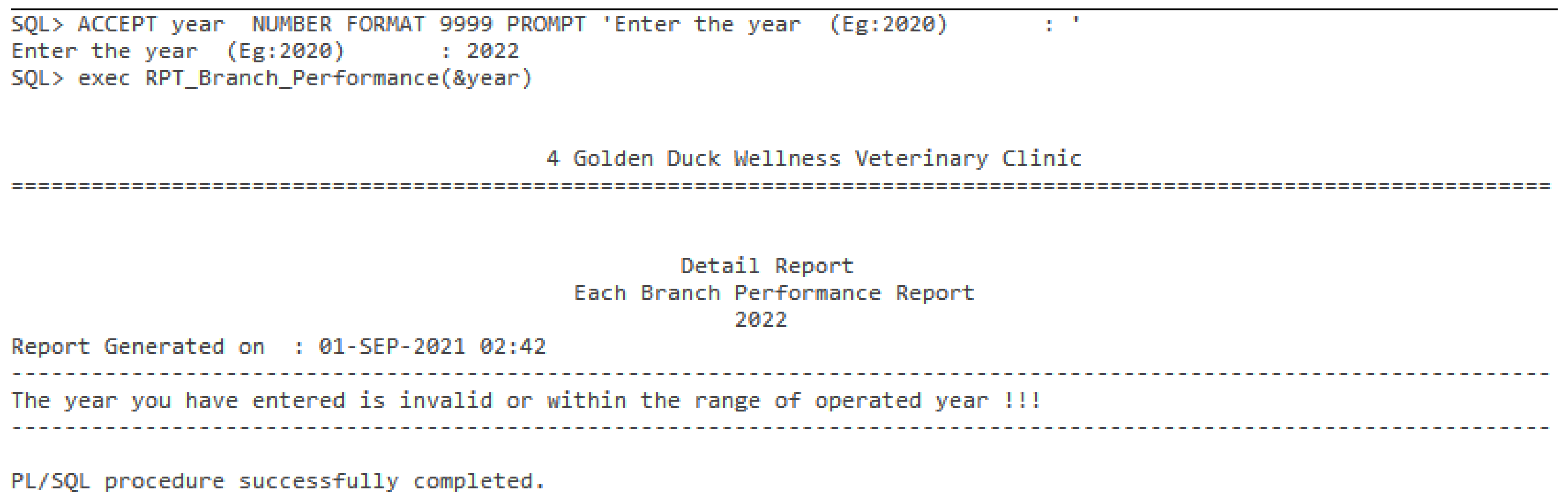


```
=============================================================================================================================
Branch ID   :B0003
State       :Kedah
City        :Alor Setar
Postcode    :5460
Street Name :11, Jalan Teluk Wanjah
-----------------------------------------------------------------------------------------------------------------------------
                                                   Revenue for each treatment
-----------------------------------------------------------------------------------------------------------------------------
Treatment ID | Treatment Type        | Treatment Price (RM) | No Treatment |     Treatment Revenue (RM) |     Percent%
-----------------------------------------------------------------------------------------------------------------------------
T0001        | Skin Care             |               150.00 |          232 |                   34800.00 |        13.21
T0002        | Dental Treatment      |               200.00 |          271 |                   54200.00 |        20.58
T0003        | Pet Emergency Care    |               400.00 |          249 |                   99600.00 |        37.81
T0004        | Gastroenteritis Care  |               180.00 |          208 |                   37440.00 |        14.21
T0005        | Antibiotics Vaccinat  |               150.00 |          249 |                   37350.00 |        14.18
-----------------------------------------------------------------------------------------------------------------------------
Total (RM)                                                                                263390.00
-----------------------------------------------------------------------------------------------------------------------------
                                                    Revenue for each Medicine
-----------------------------------------------------------------------------------------------------------------------------
Medicine ID  | Medicine Name         |  Medicine Price (RM) |     QTY Sold |      Medicine Revenue (RM) |     Percent%
-----------------------------------------------------------------------------------------------------------------------------
M0001        | Antibiotics           |                59.90 |          383 |                   22941.70 |         6.68
M0002        | Painkillers           |                79.90 |          770 |                   61523.00 |        17.91
M0003        | Multivitamins         |                89.90 |          384 |                   34521.60 |        10.05
M0004        | Probiotics            |                59.90 |          694 |                   41570.60 |         12.1
M0005        | Antioxidants          |               149.90 |          323 |                   48417.70 |        14.09
M0006        | Anthelmintics         |               200.00 |          308 |                   61600.00 |        17.93
M0007        | Skin Care Lotion      |                79.90 |          350 |                   27965.00 |         8.14
M0008        | Omega-3 fatty acids   |               125.00 |          360 |                   45000.00 |         13.1
-----------------------------------------------------------------------------------------------------------------------------
Total (RM)                                                                                343539.60
-----------------------------------------------------------------------------------------------------------------------------
```



```
--------------------------------------------------------------------------------------------------------------
                                     Contribution of each branch in 2020
--------------------------------------------------------------------------------------------------------------
Branch ID  |    Treatment Revenue (RM) |     Medicine Revenue (RM) | Branch Total Revenue (RM) |     Contribution%
--------------------------------------------------------------------------------------------------------------
B0001      |                 386250.00 |                 520923.50 |                 907173.50 |             37.46
B0002      |                 383370.00 |                 524129.20 |                 907499.20 |             37.48
B0003      |                 263390.00 |                 343539.60 |                 606929.60 |             25.06
--------------------------------------------------------------------------------------------------------------
Total (RM) |                1033010.00 |                1388592.30 |                2421602.30 |
--------------------------------------------------------------------------------------------------------------
```

Exception Output : Invalid year and out of clinic operation year



```
SQL> ACCEPT year  NUMBER FORMAT 9999 PROMPT 'Enter the year  (Eg:2020)      : '
Enter the year  (Eg:2020)      : 2022
SQL> exec RPT_Branch_Performance(&year)


                                        4 Golden Duck Wellness Veterinary Clinic
==================================================================================================================


                                                  Detail Report
                                          Each Branch Performance Report
                                                      2022
Report Generated on   : 01-SEP-2021 02:42
------------------------------------------------------------------------------------------------------------------
The year you have entered is invalid or within the range of operated year !!!
------------------------------------------------------------------------------------------------------------------

PL/SQL procedure successfully completed.
```

Chapter 5 Extra Effort Highlights

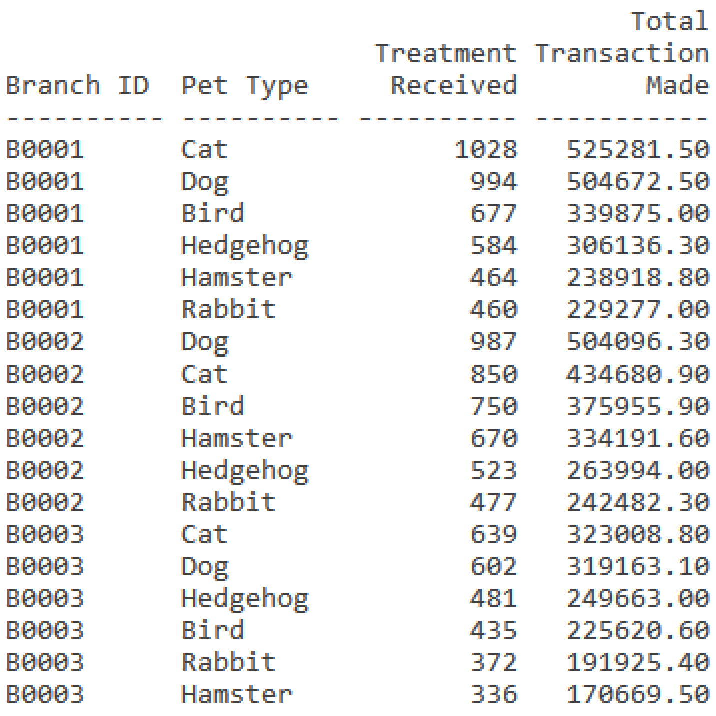
5.1 (Tan Yi Hong)

5.1.1 Views

View 1: The purpose of this view is to display the top pet type that received treatment in each branch

```
CREATE OR REPLACE VIEW topPetTreatment AS
SELECT t.branch_id, pt.type_name, COUNT(t.appointment_id) AS
NoOfTreatment, SUM(t.total_amount) AS TransactionAmount
FROM appointment a, veterinarian v, pet p, petType pt,
transaction t
WHERE t.appointment_id=a.appointment_id AND a.vet_id=v.vet_id
AND a.pet_id=p.pet_id
      AND p.type_id=pt.type_id AND
t.appointment_id=a.appointment_id
GROUP BY t.branch_id, pt.type_name
ORDER BY t.branch_id, SUM(t.total_amount) DESC;
```

Sample output :



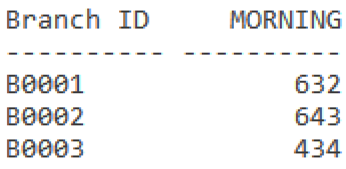
```
                                        Total
                       Treatment  Transaction
Branch ID  Pet Type     Received         Made
---------- ---------- ---------- -----------
B0001      Cat              1028   525281.50
B0001      Dog               994   504672.50
B0001      Bird              677   339875.00
B0001      Hedgehog          584   306136.30
B0001      Hamster           464   238918.80
B0001      Rabbit            460   229277.00
B0002      Dog               987   504096.30
B0002      Cat               850   434680.90
B0002      Bird              750   375955.90
B0002      Hamster           670   334191.60
B0002      Hedgehog          523   263994.00
B0002      Rabbit            477   242482.30
B0003      Cat               639   323008.80
B0003      Dog               602   319163.10
B0003      Hedgehog          481   249663.00
B0003      Bird              435   225620.60
B0003      Rabbit            372   191925.40
B0003      Hamster           336   170669.50
```

View 2: The purpose of this view is to display the amount of appointments made during morning times of each branch in last year

```
CREATE OR REPLACE VIEW morningApp AS
SELECT t.branch_id, COUNT(t.transaction_id) AS MORNING
FROM appointment a, transaction t
WHERE t.appointment_id = a.appointment_id AND
      EXTRACT(HOUR FROM CAST(a.appointment_datetime AS
TIMESTAMP)) BETWEEN 10 AND 12
```

```
              AND EXTRACT(YEAR FROM a.appointment_datetime) =
EXTRACT(YEAR FROM SYSDATE)-1
GROUP BY t.branch_id
ORDER BY t.branch_id;
```

Sample output :

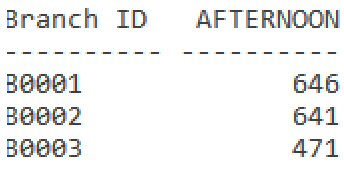


```
Branch ID     MORNING
---------- ----------
B0001             632
B0002             643
B0003             434
```

View 3: The purpose of this view is to display the amount of appointments made during afternoon times of each branch in last year

```
CREATE OR REPLACE VIEW afternoonApp AS
SELECT t.branch_id, COUNT(t.transaction_id) AS AFTERNOON
FROM appointment a, transaction t
WHERE t.appointment_id = a.appointment_id AND
      EXTRACT(HOUR FROM CAST(a.appointment_datetime AS
TIMESTAMP)) BETWEEN 13 AND 15
              AND EXTRACT(YEAR FROM a.appointment_datetime) =
EXTRACT(YEAR FROM SYSDATE)-1
GROUP BY t.branch_id
ORDER BY t.branch_id;
```

Sample output :

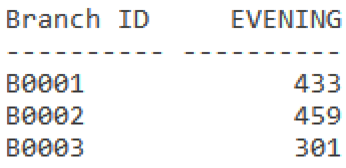


```
Branch ID   AFTERNOON
---------- ----------
B0001             646
B0002             641
B0003             471
```

View 4: The purpose of this view is to display the amount of appointments made during evening times of each branch in last year

```
CREATE OR REPLACE VIEW eveningApp AS
SELECT t.branch_id, COUNT(t.transaction_id) AS EVENING
FROM appointment a, transaction t
WHERE t.appointment_id = a.appointment_id AND
      EXTRACT(HOUR FROM CAST(a.appointment_datetime AS
TIMESTAMP)) BETWEEN 16 AND 18
              AND EXTRACT(YEAR FROM a.appointment_datetime) =
EXTRACT(YEAR FROM SYSDATE)-1
GROUP BY t.branch_id
ORDER BY t.branch_id;
```

Sample output :

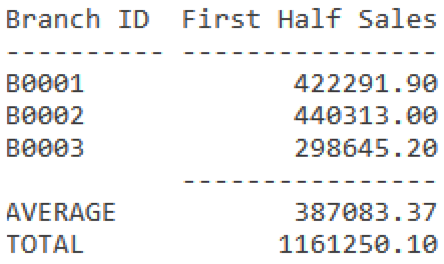


```
Branch ID     EVENING
---------- ----------
B0001             433
B0002             459
B0003             301
```

View 5: This purpose of this view is to calculate the sales of each branch in the year 2020 first half

```
CREATE OR REPLACE VIEW Sales2020_1stHalf AS
SELECT branch_id, SUM(total_amount) AS Sales2020_1stHalf
FROM transaction
WHERE EXTRACT(YEAR FROM transaction_dateTime) = 2020 AND
EXTRACT(MONTH FROM transaction_dateTime) <= 6
GROUP BY branch_id
ORDER BY branch_id, SUM(total_amount) DESC;
```

Sample output :

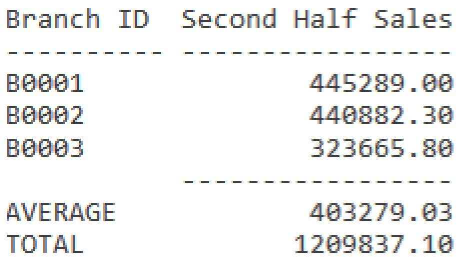


```
Branch ID  First Half Sales
---------- ----------------
B0001            422291.90
B0002            440313.00
B0003            298645.20
           ----------------
AVERAGE          387083.37
TOTAL           1161250.10
```

View 6: This purpose of this view is to calculate the sales of each branch in the year 2020 second half

```
CREATE OR REPLACE VIEW Sales2020_2ndHalf AS
SELECT branch_id, SUM(total_amount) AS Sales2020_2ndHalf
FROM transaction
WHERE EXTRACT(YEAR FROM transaction_dateTime) = 2020 AND
EXTRACT(MONTH FROM transaction_dateTime) > 6
GROUP BY branch_id
ORDER BY branch_id, SUM(total_amount) DESC;
```

Sample output :



```
Branch ID  Second Half Sales
---------- -----------------
B0001             445289.00
B0002             440882.30
B0003             323665.80
           -----------------
AVERAGE           403279.03
TOTAL            1209837.10
```

5.1.2 User Defined Exceptions

Exception 1: This exception is defined in the procedure add appointment and it will be raised when treatment ID enter by user is not found or invalid

```
   e_invalid_treatment EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_invalid_treatment, -20050);
   RAISE_APPLICATION_ERROR(-20050, 'Invalid Treatment ID.');
```

Exception 2: This exception is defined in the procedure add appointment and it will be raised when pet ID enter by user is not found or invalid

```
   e_invalid_pet EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_invalid_pet, -20051);
   RAISE_APPLICATION_ERROR(-20051, 'Invalid Pet ID.');
```

Exception 3: This exception is defined in the summary, detail, and on demand report. It will be raised when the record of report generate by user was not found

```
   e_norecord    EXCEPTION;
   PRAGMA EXCEPTION_INIT(e_norecord,-20060);
   RAISE_APPLICATION_ERROR(-20060,'No record found');
```

Exception 4: This exception is defined in trigger appointment date time and it will be raised when the appointment insert is before now.

```
 RAISE_APPLICATION_ERROR(-20052, 'Cannot insert the date time
before now.' );
```

Exception 5: This exception is defined in trigger appointment date time and it will be raised when the appointment insert is not within business hours.

```
  RAISE_APPLICATION_ERROR(-20053, 'Date time must be within
business hour.' );
```

Exception 6: This exception is defined in trigger delete appointment and it will be raised when the appointment is recorded in transaction and unable to delete

```
 RAISE_APPLICATION_ERROR(-20055,'Appointment delete
unsuccessful');
```

5.1.3 Sequences

Sequence 1: This sequence will automatically generate the appointment ID for the Kuala Lumpur branch. It will be used when inserting a new appointment in the KL branch and adding 1 after the next new appointment.

```
CREATE SEQUENCE app_seq_KL
   MINVALUE 10000001
   MAXVALUE 99999999
   START WITH  10000001
   INCREMENT BY 1;
```

Sequence 2: This sequence will automatically generate the appointment ID for the Pulau Pinang branch. It will be used when inserting a new appointment in the PG branch and adding 1 after the next new appointment.

```
CREATE SEQUENCE app_seq_PG
   MINVALUE 10000001
   MAXVALUE 99999999
   START WITH  10000001
   INCREMENT BY 1;
```

Sequence 3: This sequence will automatically generate the appointment ID for the Kedah branch. It will be used when inserting a new appointment in the KD branch and adding 1 after the next new appointment.

```
CREATE SEQUENCE app_seq_KD
   MINVALUE 10000001
   MAXVALUE 99999999
   START WITH  10000001
   INCREMENT BY 1;
```

5.1.4 Triggers

Trigger 1: This trigger is use to validate the age of the veterinarian that should be above 22 years old when a new veterinarian is inserted into the database

```
CREATE OR REPLACE TRIGGER trgVetAge
  BEFORE INSERT OR UPDATE ON Veterinarian
  FOR EACH ROW
BEGIN
  IF((ROUND((SYSDATE-:new.vet_dob)/365)) < 22) THEN
    RAISE_APPLICATION_ERROR(-20002, 'Veterinarian must be at
least 22 years old.' );
  END IF;
END;
/
```

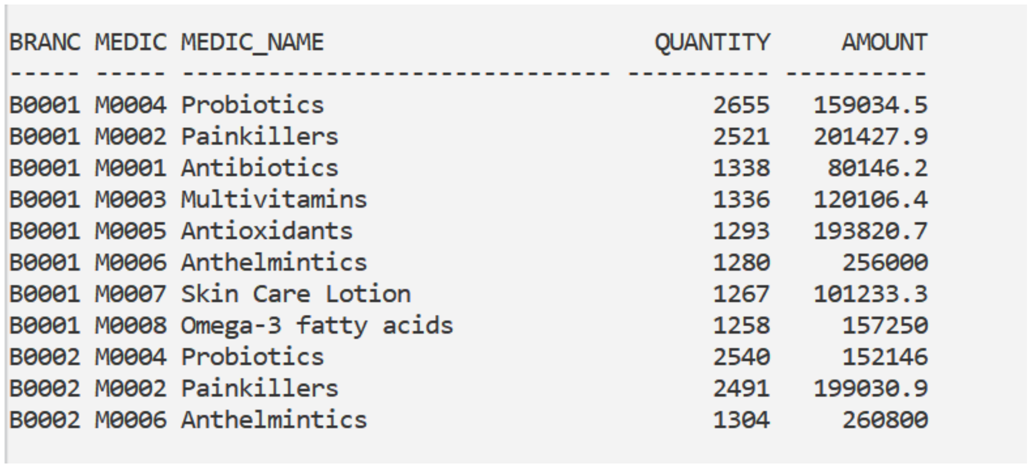
5.2 (Tan Teoh Xin Ee)

5.2.1 Views

View 1: The purpose of this view is to store the information of medicine details from different branches. For example, the medicine name, qty and amount.

```
create or replace view medicalUsed As
select t.branch_id, d.medic_id, m.medic_name, sum(d.line_qty)
as quantity, sum(d.line_total) as amount
from transaction t, branch b, transactiondetail d,
medicalsupply m
where b.branch_id = t.branch_id
      and t.transaction_id = d.transaction_id
      and d.medic_id = m.medic_id
group by t.branch_id, d.medic_id,m.medic_name
order by t.branch_id,sum(d.line_qty) DESC;
```

Sample Output:



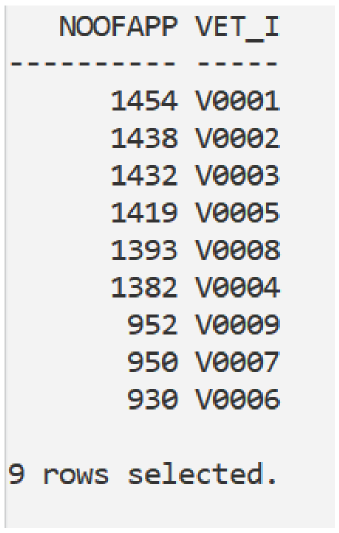
```
BRANC MEDIC MEDIC_NAME                       QUANTITY     AMOUNT
----- ----- ------------------------------ ---------- ----------
B0001 M0004 Probiotics                           2655   159034.5
B0001 M0002 Painkillers                          2521   201427.9
B0001 M0001 Antibiotics                          1338    80146.2
B0001 M0003 Multivitamins                        1336   120106.4
B0001 M0005 Antioxidants                         1293   193820.7
B0001 M0006 Anthelmintics                        1280     256000
B0001 M0007 Skin Care Lotion                     1267   101233.3
B0001 M0008 Omega-3 fatty acids                  1258     157250
B0002 M0004 Probiotics                           2540     152146
B0002 M0002 Painkillers                          2491   199030.9
B0002 M0006 Anthelmintics                        1304     260800
```

View 2: The purpose of this view is to store the numbers of appointments received by every veterinarian.

```
create or replace view appointNum As
```

```
select count(appointment_id) as NoOfapp, vet_id
from appointment
group by vet_id
order by count(appointment_id) desc;
```

Sample Output:

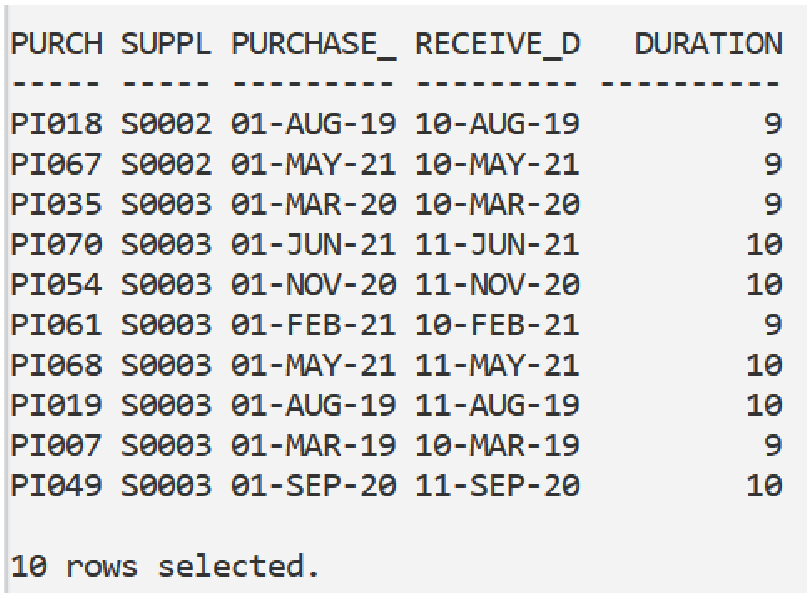


```
   NOOFAPP VET_I
---------- -----
      1454 V0001
      1438 V0002
      1432 V0003
      1419 V0005
      1393 V0008
      1382 V0004
       952 V0009
       950 V0007
       930 V0006

9 rows selected.
```

View 3: The purpose of this view is to store late sent stock information from the supplier such as purchase date, receive date and the duration from purchase date until receive date. The stock normally will be sent within 6 days.

```
create or replace view difdate as
select purchase_id, supplier_id, purchase_date,
receive_date,(receive_date-purchase_date) as duration
from purchaseTransaction
where receive_date-purchase_date>6
group by purchase_id,supplier_id, purchase_date, receive_date
order by supplier_id;
```

Sample Output:



```
PURCH SUPPL PURCHASE_ RECEIVE_D   DURATION
----- ----- --------- --------- ----------
PI018 S0002 01-AUG-19 10-AUG-19          9
PI067 S0002 01-MAY-21 10-MAY-21          9
PI035 S0003 01-MAR-20 10-MAR-20          9
PI070 S0003 01-JUN-21 11-JUN-21         10
PI054 S0003 01-NOV-20 11-NOV-20         10
PI061 S0003 01-FEB-21 10-FEB-21          9
PI068 S0003 01-MAY-21 11-MAY-21         10
PI019 S0003 01-AUG-19 11-AUG-19         10
PI007 S0003 01-MAR-19 10-MAR-19          9
PI049 S0003 01-SEP-20 11-SEP-20         10

10 rows selected.
```

5.2.2 User Defined Exceptions

The purpose of this exception is to prompt the user that 'Invalid supplier code.', if he/she key in the wrong supplier id.

```
EXCE_SUPPLIERCODE EXCEPTION;
PRAGMA EXCEPTION_INIT(EXCE_SUPPLIERCODE, -20310);
```

5.2.3 Sequence

The purpose of this sequence is to generate numbers for medic_id, which will be needed in the medic add procedure.

```
CREATE SEQUENCE MEDICID
  MINVALUE 8
  MAXVALUE 9999
  START WITH 8
  INCREMENT BY 1;
```

5.2.4 Trigger

The purpose of this trigger is to delete the amount of purchase items from the amount of purchase transaction, if the purchase item was deleted.

```
CREATE OR REPLACE TRIGGER trgDelPurchaseItem
  After Delete ON PurchaseItem
  FOR EACH ROW
 BEGIN
  Update PurchaseTransaction
   SET purchase_amount = purchase_amount - (:new.purchase_qty
* :new.purchase_price)
   where purchase_id = :new.purchase_id;
END;
/
```

5.3 (Tan Wei Siong)

5.3.1 User Defined Exceptions

Exception 1: This exception is defined in the trigger check appointment date time to check whether the selected time for the veterinarian is book or not. If the time has been booked, the exception will raise and output the suggested time for the user.

```
   Date_Time_Booked EXCEPTION;
   PRAGMA exception_init(Date_Time_Booked, -20200 );
```

Exception 2: This exception is defined in the procedure pet register to check whether the provided owner information exists or not. It will raise when the owner information is not found.

```
No_owner_found EXCEPTION;
 PRAGMA exception_init(No_owner_found,-20201);
```

Exception 3: This exception is defined in the on-demand report of the pet treatment detail. The exception will raise when the user enters the invalid pet id into the system. It will message the user that the pet is not found.

```
 NO_PET_FOUND EXCEPTION;
 PRAGMA EXCEPTION_INIT(NO_PET_FOUND, -20202);
```

Exception 4: This exception is defined in the trigger check owner age. The exception will raise when the registered user age is below 18.

```
RAISE_APPLICATION_ERROR(-20004, 'Pet Owner must be at least
18 years old.' );
```

5.3.2 Sequence

Sequence 1: This sequence is used to auto generate the pet owner id. The sequence number will increase 1 when there is a new pet owner registered.

```
CREATE SEQUENCE owner_seq
 START WITH 501
 INCREMENT BY 1;
```

Sequence 2: This sequence is used to auto generate the pet id. The sequence number will increase 1 when there is a new pet registered.

```
CREATE SEQUENCE pet_seq
 START WITH 1001
 INCREMENT BY 1;
```

5.3.3 Procedure

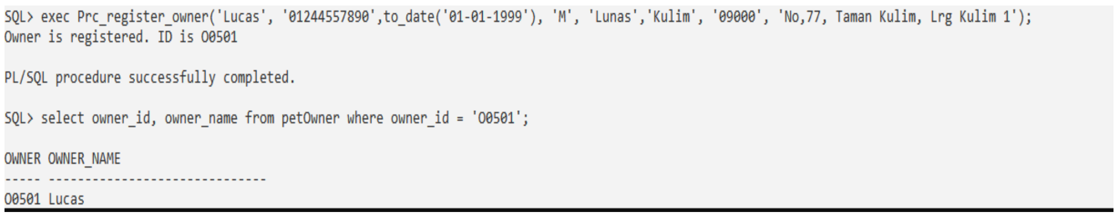
5.3.3.1 Procedure 1: Pet Owner Registration

Purpose: The purpose of this procedure is to let the staff register the pet owner in an easier way. The staff online need to input the owner information into this procedure and the owner will be added into the database.

Procedure code:

```
CREATE OR REPLACE Procedure
Prc_register_owner(in_owner_Name IN VARCHAR2,
in_owner_Contact IN VARCHAR2,

in_owner_dob IN Date, in_gender IN CHAR, in_state IN
VARCHAR2,

           in_city IN VARCHAR2, in_postcode IN VARCHAR2,

                          in_streetName IN VARCHAR2) AS
 v_owner_seq VARCHAR2(4);
 v_petOwner_id PetOwner.owner_id%TYPE;

BEGIN
 SELECT to_char(lpad(owner_seq.nextval,'4','0')) INTO
v_owner_seq from dual;
 v_petOwner_id := ('O'||v_owner_seq);
 Insert into petOwner values(v_petOwner_id,
in_owner_Name , in_owner_Contact, in_owner_dob,
                             in_gender, in_state,
in_city, in_postcode, in_streetName);

dbms_output.put_line('Owner is registered. ID is ' ||
v_petOwner_id);
END;
/
```

Sample Output:



```
SQL> exec Prc_register_owner('Lucas', '01244557890',to_date('01-01-1999'), 'M', 'Lunas','Kulim', '09000', 'No,77, Taman Kulim, Lrg Kulim 1');
Owner is registered. ID is O0501

PL/SQL procedure successfully completed.

SQL> select owner_id, owner_name from petOwner where owner_id = 'O0501';

OWNER OWNER_NAME
----- ------------------------------
O0501 Lucas
```

5.3.4 Trigger

Trigger 1: The purpose of this trigger is to auto update the purchase amount when there is a new purchase item inserted into the purchase item table.

```
CREATE OR REPLACE TRIGGER trgPurchaseItem
  After Insert ON PurchaseItem
  FOR EACH ROW
 BEGIN
  Update PurchaseTransaction
   SET purchase_amount = purchase_amount + (:new.purchase_qty
* :new.purchase_price)
   where purchase_id = :new.purchase_id;
END;
/
```

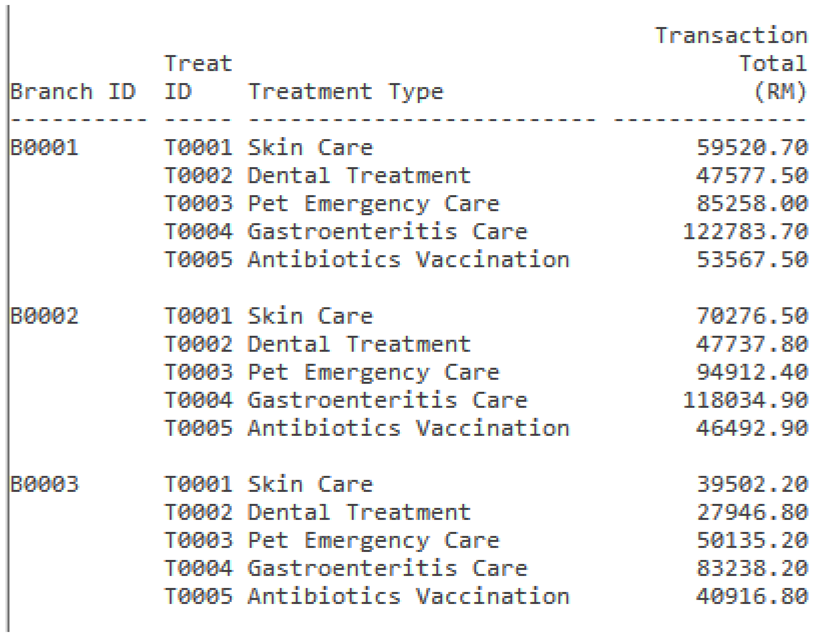
5.4 (Nigel Lee Jian Hsee)

5.4.1 Views

View 1: The purpose of this view is to store the treatment revenue in the year of 2021 of each branch.

```
CREATE OR REPLACE VIEW FullRevenueGroupByTreatment2021
AS
select  T.branch_id,TT.treatment_id, TT.treatment_type,
sum(T.total_amount) as Total_Amount
from transaction T, appointment A, treatment TT
where A.appointment_id = T.appointment_id AND
A.treatment_id = TT.treatment_id AND EXTRACT(YEAR FROM
T.transaction_datetime) = '2021'
group by T.branch_id,TT.treatment_id, TT.treatment_type

order by branch_id;
```

Sample Output:

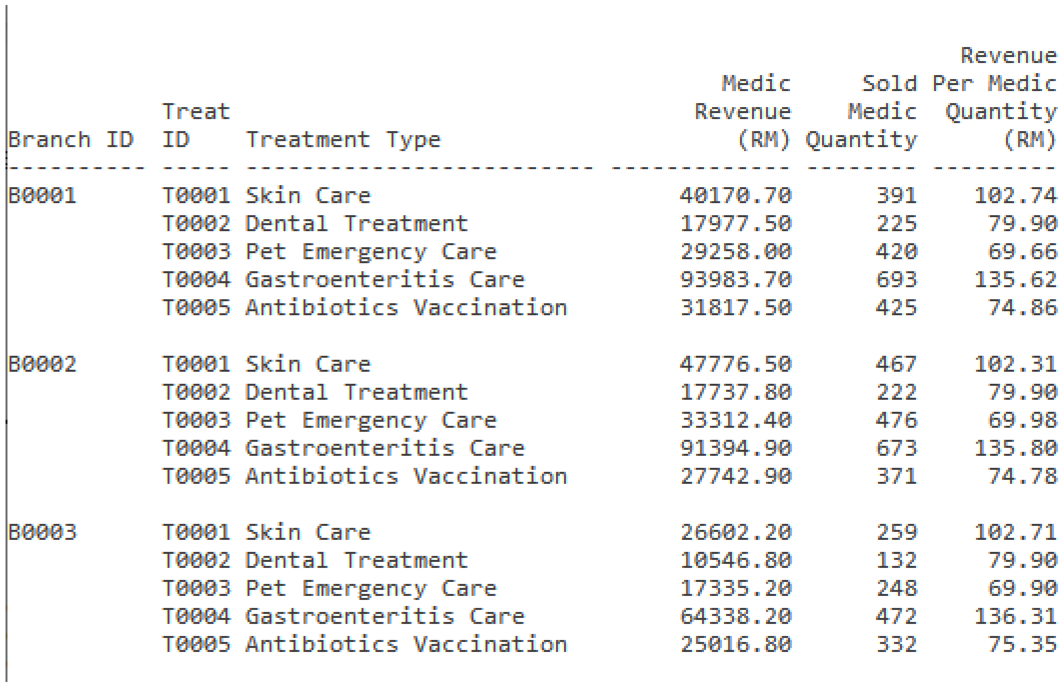


```
                                                        Transaction
           Treat                                              Total
Branch ID  ID    Treatment Type                                (RM)
---------- ----- ------------------------- --------------
B0001      T0001 Skin Care                         59520.70
           T0002 Dental Treatment                  47577.50
           T0003 Pet Emergency Care                85258.00
           T0004 Gastroenteritis Care             122783.70
           T0005 Antibiotics Vaccination           53567.50

B0002      T0001 Skin Care                         70276.50
           T0002 Dental Treatment                  47737.80
           T0003 Pet Emergency Care                94912.40
           T0004 Gastroenteritis Care             118034.90
           T0005 Antibiotics Vaccination           46492.90

B0003      T0001 Skin Care                         39502.20
           T0002 Dental Treatment                  27946.80
           T0003 Pet Emergency Care                50135.20
           T0004 Gastroenteritis Care              83238.20
           T0005 Antibiotics Vaccination           40916.80
```

View 1: The purpose of this view is to store the medicine revenue of each treatment in the year of 2021 of each branch. For example, total medicine revenue that earned after a dental treatment.

```
CREATE OR REPLACE VIEW MedicRevenueGroupByTreatment2021
AS
select T.branch_id,TT.treatment_id, TT.treatment_type,
sum(TD.line_total) AS Medic_Revenue, sum(TD.line_qty) AS
Sold_Quantity, (sum(TD.line_total)/sum(TD.line_qty)) AS
Revenue_Per_Quantity
from transactiondetail TD, transaction T, appointment A,
treatment TT
where TD.transaction_id = T.transaction_id AND
A.appointment_id = T.appointment_id AND A.treatment_id =
TT.treatment_id AND EXTRACT(YEAR FROM
T.transaction_datetime) = '2021'
group by T.branch_id,TT.treatment_id, TT.treatment_type
order by branch_id;
```

Sample Output:



```
                                                                         Revenue
                                                          Medic     Sold Per Medic
           Treat                                        Revenue    Medic  Quantity
Branch ID  ID    Treatment Type                            (RM) Quantity      (RM)
---------- ----- ------------------------- ------------- -------- ---------
B0001      T0001 Skin Care                         40170.70      391    102.74
           T0002 Dental Treatment                  17977.50      225     79.90
           T0003 Pet Emergency Care                29258.00      420     69.66
           T0004 Gastroenteritis Care              93983.70      693    135.62
           T0005 Antibiotics Vaccination           31817.50      425     74.86

B0002      T0001 Skin Care                         47776.50      467    102.31
           T0002 Dental Treatment                  17737.80      222     79.90
           T0003 Pet Emergency Care                33312.40      476     69.98
           T0004 Gastroenteritis Care              91394.90      673    135.80
           T0005 Antibiotics Vaccination           27742.90      371     74.78

B0003      T0001 Skin Care                         26602.20      259    102.71
           T0002 Dental Treatment                  10546.80      132     79.90
           T0003 Pet Emergency Care                17335.20      248     69.90
           T0004 Gastroenteritis Care              64338.20      472    136.31
           T0005 Antibiotics Vaccination           25016.80      332     75.35
```

5.4.2 User Defined Exceptions

Exception 1: This exception is defined in the procedure add and create transaction to check whether the entered appointment exists. If not an error message will be displayed.

```
  e_invalid_appointmentid EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_invalid_appointmentid, -20150);
```

Exception 2: This exception is defined in the procedure add and create transaction to check whether the entered appointment id transaction is already created . If yes an error message will be displayed.

```
  e_repeated_transaction EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_repeated_transaction, -20151);
```

Exception 3: This exception is defined in the adding transaction detail procedure. The exception will raise when the user adds a medicine that already exists in a created transaction. If yes an error message will be prompted.

```
  e_repeated_medicid EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_repeated_medicid, -20155);
```

Exception 4: This exception is defined in the adding transaction detail procedure. The exception will raise when the medic quantity entered is less than or equal zero.

```
  e_zero_qty EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_zero_qty, -20156);
```

Exception 5: This exception is defined in the edit transaction detail procedure. The exception will raise when the transaction is an old transaction that has already passed 7 days.

```
  e_dayexceed EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_dayexceed, -20157)
```

Exception 5: This exception is defined in the edit transaction detail procedure. The exception will raise when the user enters a transaction detail that does not exist.

```
  e_invalid_transactiondetail EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_invalid_transactiondetail, -20158);
```

Exception 6: This exception is defined in the edit transaction detail procedure. The exception will raise when the quantity is less than zero.

```
  e_lesszero_qty EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_lesszero_qty, -20159);
```

Exception 7: This exception is defined in the edit transaction detail procedure. The exception will raise when the medicine quantity entered is the same as the old quantity and remind the user that the quantity is the same and will not change.

```
  e_samequantity EXCEPTION;
  PRAGMA EXCEPTION_INIT(e_samequantity, -20161);
```

Exception 8: This exception is defined in the vet monthly performance summary report. The exception will raise when the entered vet id is invalid.

```
 e_invalid_vetid EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_vetid, -20162);
```

Exception 9: This exception is defined in the vet monthly performance summary report. The exception will raise when the entered month is invalid.

```
 e_invalid_month EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_month, -20163);
```

Exception 10: This exception is defined in the branch performance yearly detail report and in the vet monthly performance summary report. The exception will raise when the year is invalid.

```
 e_invalid_year EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_year, -20164);
```

Exception 11: This exception is defined in the branch poor performance on demand report. The exception will raise when the entered branch id is invalid.

```
 e_invalid_branch EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_branch, -20165);
```

Exception 12: This exception is defined in the branch performance yearly detail report. The exception will raise when the entered year is inavlid and out of range.

```
 e_invalid_year EXCEPTION;
 PRAGMA EXCEPTION_INIT(e_invalid_year, -20166);
```

5.4.3 Trigger

Trigger 1: The purpose of this trigger is to validate pet age by checking the date of birth. Which means when the date of birth is after the system, an error will be raised to prompt the user. The pet age must be more than 1.

```
CREATE OR REPLACE TRIGGER trgPetAge
  BEFORE INSERT OR UPDATE ON Pet
  FOR EACH ROW
BEGIN
  IF((ROUND((SYSDATE-:new.pet_dob)/365)) < 0) THEN
    RAISE_APPLICATION_ERROR(-20003, 'Pet must be at least more than
0 years old.' );
  END IF;
END;
/
```