Online Vet Appointment System

Table of Content -

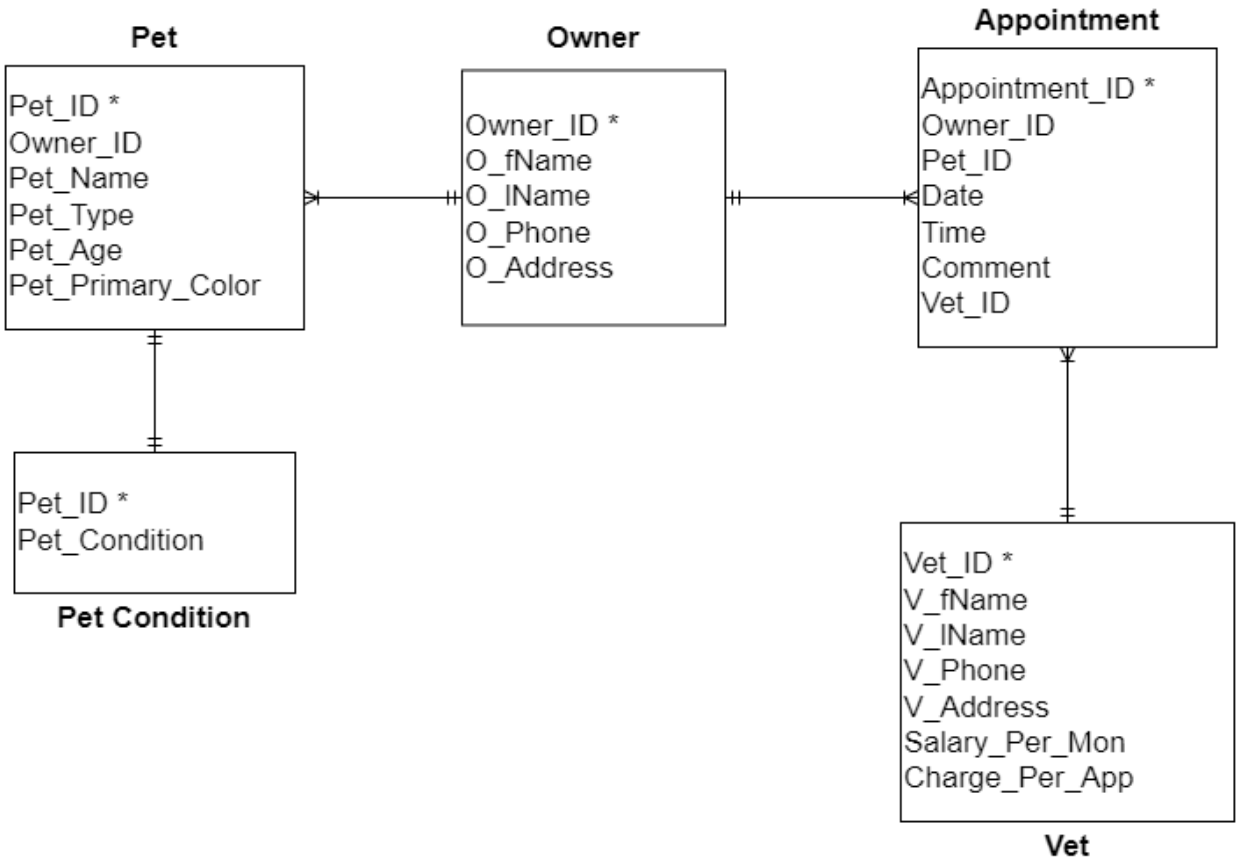
Project Scenario (Intro)

Our plan is to develop a database system for a pet clinic to give online appointments. The clinic will have vets / staff, pet owners and their pets.

Moreover, we develop a database which can store vets/staff information, pet owners information, their pets information, and appointments information. We also create functionality to our database application so that it can perform complex tasks using functions and procedures. Moreover, our database application also can add, modify, or remove entries from the system.

Finally, we also focus on our developing database system/application and its blueprint to perform tasks with real life applications / websites.

Entity Relational Diagram (ER Diagram)



Data Dictionary

Pet

ATTRIBUTE NAME	DATA TYPE	SIZE	NULL
Pet_ID	NUMBER	4	NO (PK)
Owner_ID	NUMBER	4	NO (FK)
Pet_Name	VARCHAR2	25	NO
Pet_Type	VARCHAR2	10	NO (CHECK)
Pet_Age	NUMBER	2	NO
Pet_Primary_Color	VARCHAR2	15	Yes

Pet_Condition

ATTRIBUTE NAME	DATA TYPE	SIZE	NULL
Pet_ID	NUMBER	4	NO (PK, FK)
Pet_Condition	VARCHAR2	50	YES

Owner

ATTRIBUTE NAME	DATA TYPE	SIZE	NULL
Owner_ID	NUMBER	4	NO (PK)
O_fName	VARCHAR2	20	NO
O_lName	VARCHAR2	20	NO
O_Phone	VARCHAR2	12	UNIQUE
O_Address	VARCHAR2	50	NO

Appointment

ATTRIBUTE NAME	DATA TYPE	SIZE	NULL
Appointment_ID	NUMBER	5	NO (PK)
Owner_ID	NUMBER	4	NO (FK)
Pet_ID	NUMBER	4	NO (FK)
Date	DATE		NO
Time	VARCHAR2	20	NO
Comment	VARCHAR2	40	Yes
Vet_ID	NUMBER	3	NO (FK)

Vet

ATTRIBUTE NAME	DATA TYPE	SIZE	NULL
Vet_ID	NUMBER	3	NO (PK)
V_fName	NUMBER	20	NO
V_lName	VARCHAR2	20	NO
V_Phone	VARCHAR2	12	UNIQUE
V_Address	NUMBER	50	YES
Salary_Per_Mon	NUMBER	6	NO
Charge_Per_App	NUMBER	3	NO

Create all the required tables

Drop Statement for All Table -

```
DROP TABLE Vet CASCADE CONSTRAINTS;
DROP TABLE Owner CASCADE CONSTRAINTS;
DROP TABLE Pet CASCADE CONSTRAINTS;
DROP TABLE Pet_Condition CASCADE CONSTRAINTS;
DROP TABLE Appointment CASCADE CONSTRAINTS;
```

Create **Vet** Table -

```
CREATE TABLE Vet (
   Vet_ID NUMBER(3),
   V_fName VARCHAR2(20) NOT NULL,
   V_lName VARCHAR2(20) NOT NULL,
   V_Phone VARCHAR2(12) UNIQUE,
   V_Address VARCHAR2(50),
   Salary_Per_Mon NUMBER(6) NOT NULL,
   Charge_Per_App NUMBER(3) NOT NULL,
   CONSTRAINT Vet_VetID_PK PRIMARY KEY (Vet_ID)
);
```

Create **Owner** Table-

```
CREATE TABLE Owner (
   Owner_ID NUMBER(4),
   O_fName VARCHAR2(20) NOT NULL,
   O_lName VARCHAR2(20) NOT NULL,
   O_Phone VARCHAR2(12) UNIQUE,
   O_Address VARCHAR2(50) NOT NULL,
   CONSTRAINT Owner_OwnerID_PK PRIMARY KEY (Owner_ID)
);
```

Create **Pet** Table -

```
CREATE TABLE Pet (
   Pet_ID NUMBER(4),
   Owner_ID NUMBER(4),
   Pet_Name VARCHAR2(25) NOT NULL,
   Pet_Type VARCHAR2(10) CHECK(lower(Pet_Type) IN ('dog', 'cat', 'bird', 'rabbit', 'fish',
'Cow', 'Goat', 'reptile')),
   Pet_Age NUMBER(2),
   Pet_Primary_Color VARCHAR2(15),
   CONSTRAINT Pet_PetID_PK PRIMARY KEY (Pet_ID),
   CONSTRAINT Pet_OwnerID_FK FOREIGN KEY (Owner_ID) REFERENCES Owner
);
```

Create **Pet_Condition** Table -

```
CREATE TABLE Pet_Condition (
   Pet_ID NUMBER(4),
   Pet_Condition VARCHAR2(50),
   CONSTRAINT PetCondition_PetID_PK PRIMARY KEY (Pet_ID),
   CONSTRAINT PetCondition_PetID_FK FOREIGN KEY (Pet_ID) REFERENCES Pet
);
```

Create **Appointment** Table -

```
CREATE TABLE Appointment (
   Appointment_ID NUMBER(5),
   Owner_ID NUMBER(4),
   Pet_ID NUMBER(4),
   "Date" DATE NOT NULL,
   "Time" VARCHAR2(20) NOT NULL,
   "Comment" VARCHAR2(40),
   Vet_ID NUMBER(3),
   CONSTRAINT Appointment_AppID_PK PRIMARY KEY (Appointment_ID),
   CONSTRAINT Appointmemt_OwnerID_FK FOREIGN KEY (Owner_ID) REFERENCES
Owner,
   CONSTRAINT Appointmemt_PetID_FK FOREIGN KEY (Pet_ID) REFERENCES Pet,
   CONSTRAINT Appointmemt_VetID_PK FOREIGN KEY (Vet_ID) REFERENCES
);
```

Insert data into tables

Insert data into **Vet** table -

```
INSERT INTO Vet
VALUES(111, 'Tom', 'Harij', '01112223334', 'Gombak, Selangor', '6000', '80');

INSERT INTO Vet
VALUES(112, 'Mohammad', 'Imran', '01112663334', 'Bukit Damansara, Kuala Lumpur', '6500',
'90');

INSERT INTO Vet
VALUES(113, 'Nurul', 'Hushni', '01166623334', 'Medan Idaman, Selangor', '8000', '100');

INSERT INTO Vet
VALUES(114, 'Yu', 'Yan', '01112223434', 'Bukit Bintang, Kuala Lumpur', '9000', '110');

INSERT INTO Vet
VALUES(115, 'Ying', 'Yue', '01122223434', 'Bukit Bintang, Kuala Lumpur', '9000', '120');

INSERT INTO Vet
VALUES(116, 'Safa', 'Kabir', '01212223434', 'Batu Caves, Selangor', '8500', '100');

INSERT INTO Vet
VALUES(117, 'Mohammad', 'Musfiq', '01212923434', 'Batu Caves, Selangor', '7000', '70');

INSERT INTO Vet
VALUES(118, 'Mohammad', 'Latif', '01912923434', 'Ampang, Kuala Lumpur', '7500', '85');

INSERT INTO Vet
VALUES(119, 'Mohammad', 'Razak', '01219923434', 'Batu Caves, Selangor', '10000', '150');

INSERT INTO Vet
VALUES(120, 'Mohammad', 'Mubarak', '01999923434', 'Batu Caves, Selangor', '10000', '140');
```

Insert data into **Owner** table -

```
INSERT INTO Owner
VALUES(1001, 'Esmee', 'Benitez', '01100000111', 'Gombak, Selangor');

INSERT INTO Owner
```

```
VALUES(1002, 'Freyja', 'Paul', '01100000112', 'Ampang, Selangor');

INSERT INTO Owner
VALUES(1003, 'Mohammad', 'Iman', '01100000113', 'Bandar Baru Bangi, Selangor');

INSERT INTO Owner
VALUES(1004, 'Mohammad', 'Asraf', '01100000114', 'Banting, Selangor');

INSERT INTO Owner
VALUES(1005, 'Mohammad', 'Arif', '01100001114', 'Gombak, Selangor');

INSERT INTO Owner
VALUES(1006, 'Danial', 'Hakim', '01100000115', 'Medan Idaman, Selangor');

INSERT INTO Owner
VALUES(1007, 'Raihan', 'Kabir', '01100000116', 'Batu Caves, Selangor');

INSERT INTO Owner
VALUES(1008, 'Mohammad', 'Hahmid', '01100001116', 'Gombak, Selangor');

INSERT INTO Owner
VALUES(1009, 'Mohammad', 'Rakib', '01100001117', 'Bukit Nanas, Kuala Lumpur');

INSERT INTO Owner
VALUES(1010, 'Abdur', 'Rahman', '01100000118', 'Gombak, Selangor');
```

Insert data into **Owner** table -

```
INSERT INTO Owner
VALUES(1001, 'Esmee', 'Benitez', '01100000111', 'Gombak, Selangor');

INSERT INTO Owner
VALUES(1002, 'Freyja', 'Paul', '01100000112', 'Ampang, Selangor');

INSERT INTO Owner
VALUES(1003, 'Mohammad', 'Iman', '01100000113', 'Bandar Baru Bangi, Selangor');

INSERT INTO Owner
VALUES(1004, 'Mohammad', 'Asraf', '01100000114', 'Banting, Selangor');

INSERT INTO Owner
VALUES(1005, 'Mohammad', 'Arif', '01100001114', 'Gombak, Selangor');

INSERT INTO Owner
```

```
VALUES(1006, 'Danial', 'Hakim', '01100000115', 'Medan Idaman, Selangor');

INSERT INTO Owner
VALUES(1007, 'Raihan', 'Kabir', '01100000116', 'Batu Caves, Selangor');

INSERT INTO Owner
VALUES(1008, 'Mohammad', 'Hahmid', '01100001116', 'Gombak, Selangor');

INSERT INTO Owner
VALUES(1009, 'Mohammad', 'Rakib', '01100001117', 'Bukit Nanas, Kuala Lumpur');

INSERT INTO Owner
VALUES(1010, 'Abdur', 'Rahman', '01100000118', 'Gombak, Selangor');

INSERT INTO Owner
VALUES(1011, 'Abdur', 'Salem', '01100010118', ' ');
```

Insert data into **Pet** table -

```
INSERT INTO Pet
VALUES(1101, 1001, 'Mini', 'cat', 2, 'white');

INSERT INTO Pet
VALUES(1102, 1002, 'Luna', 'cat', 5, 'brown');

INSERT INTO Pet
VALUES(1103, 1002, 'Teddy', 'dog', 7, 'orange');

INSERT INTO Pet
VALUES(1104, 1003, 'Bill', 'dog', 4, 'black');

INSERT INTO Pet
VALUES(1105, 1004, 'Bella', 'dog', 5, 'brown');

INSERT INTO Pet
VALUES(1106, 1004, 'Buddy', 'dog', 2, 'white');

INSERT INTO Pet
VALUES(1107, 1008, 'Milo', 'cat', 1, 'orange');

INSERT INTO Pet
VALUES(1108, 1005, 'Loki', 'cat', 2, 'black');

INSERT INTO Pet
```

```
VALUES(1109, 1006, 'Oliver', 'cat', 5, 'white');

INSERT INTO Pet
VALUES(1110, 1007, 'Bubba', 'bird', 1, 'green');

INSERT INTO Pet
VALUES(1111, 1009, 'Baldy', 'cat', 3, 'black');

INSERT INTO Pet
VALUES(1112, 1010, 'Putih', 'cat', 5, 'black');
```

Insert data into **Pet_Condition** table -

```
INSERT INTO Pet_Condition
VALUES(1101, 'Good health');

INSERT INTO Pet_Condition
VALUES(1102, 'Not good');

INSERT INTO Pet_Condition
VALUES(1104, 'Ringworm');

INSERT INTO Pet_Condition
VALUES(1107, 'Psittacosis');

INSERT INTO Pet_Condition
VALUES(1112, 'Diabetes');

INSERT INTO Pet_Condition
VALUES(1111, 'Heartworm');
```

Insert data into **Appointment** table -

```
INSERT INTO Appointment
VALUES(10001, 1001, 1101, to_date('11/07/22', 'dd/mm/yy'), '11.00 am', null, 111);

INSERT INTO Appointment
VALUES(10002, 1002, 1102, to_date('14/07/22', 'dd/mm/yy'), '10.00 am', 'Rash on Skin', 111);
```

```
INSERT INTO Appointment
VALUES(10003, 1003, 1104, to_date('09/07/22', 'dd/mm/yy'), '12.00 pm', null, 112);

INSERT INTO Appointment
VALUES(10004, 1004, 1105, to_date('14/07/22', 'dd/mm/yy'), '05.00 pm', null, 114);

INSERT INTO Appointment
VALUES(10005, 1006, 1109, to_date('18/07/22', 'dd/mm/yy'), '10.00 am', 'Rash on Skin', 119);

INSERT INTO Appointment
VALUES(10006, 1010, 1112, to_date('20/07/22', 'dd/mm/yy'), '11.00 am', null, 120);

INSERT INTO Appointment
VALUES(10007, 1009, 1111, to_date('15/07/22', 'dd/mm/yy'), '09.00 am', null, 118);

INSERT INTO Appointment
VALUES(10008, 1008, 1107, to_date('13/07/22', 'dd/mm/yy'), '01.00 pm', null, 112);

INSERT INTO Appointment
VALUES(10009, 1002, 1103, to_date('14/07/22', 'dd/mm/yy'), '10.00 am', null, 111);

INSERT INTO Appointment
VALUES(10010, 1005, 1108, to_date('10/07/22', 'dd/mm/yy'), '11.00 am', 'Diabetes', 114);
```

SQL Statements

- **List all pet owner orders by their id.**

```
SELECT *
FROM owner
ORDER BY owner_id;
```

0.078 seconds

Worksheet | Query Builder

```
-- List all pet owner order by their id
SELECT *
FROM owner
ORDER BY owner_id;
```

Script Output

Task completed in 0.078 seconds

OWNER_ID	O_FNAME	O_LNAME	O_PHONE	O_ADDRESS
1001	Esmee	Benitez	01100000111	Gombak, Selangor
1002	Freyja	Paul	01100000112	Ampang, Selangor
1003	Mohammad	Iman	01100000113	Bandar Baru Bangi, Selangor
1004	Mohammad	Asraf	01100000114	Banting, Selangor
1005	Mohammad	Arif	01100001114	Gombak, Selangor
1006	Danial	Hakim	01100000115	Medan Idaman, Selangor
1007	Raihan	Kabir	01100000116	Batu Caves, Selangor
1008	Mohammad	Hahmid	01100001116	Gombak, Selangor
1009	Mohammad	Rakib	01100001117	Bukit Nasa, Kuala Lumpur
1010	Abdur	Rahman	01100000118	Gombak, Selangor
1011	Abdur	Salem	01100010118	

11 rows selected.

- **List all pet owners with their pet information.**

```
SELECT o.owner_id, o.o_fname, o.o_lname, p.pet_id, p.pet_name, p.pet_type
FROM owner o
JOIN pet p
ON o.owner_id = p.owner_id
ORDER BY o.owner_id;
```

0.078 seconds

Worksheet | Query Builder

```
-- Select all pet owner with their pet information
SELECT o.owner_id, o.o_fname, o.o_lname, p.pet_id, p.pet_name, p.pet_type
FROM owner o
JOIN pet p
ON o.owner_id = p.owner_id
ORDER BY o.owner_id;
```

Script Output

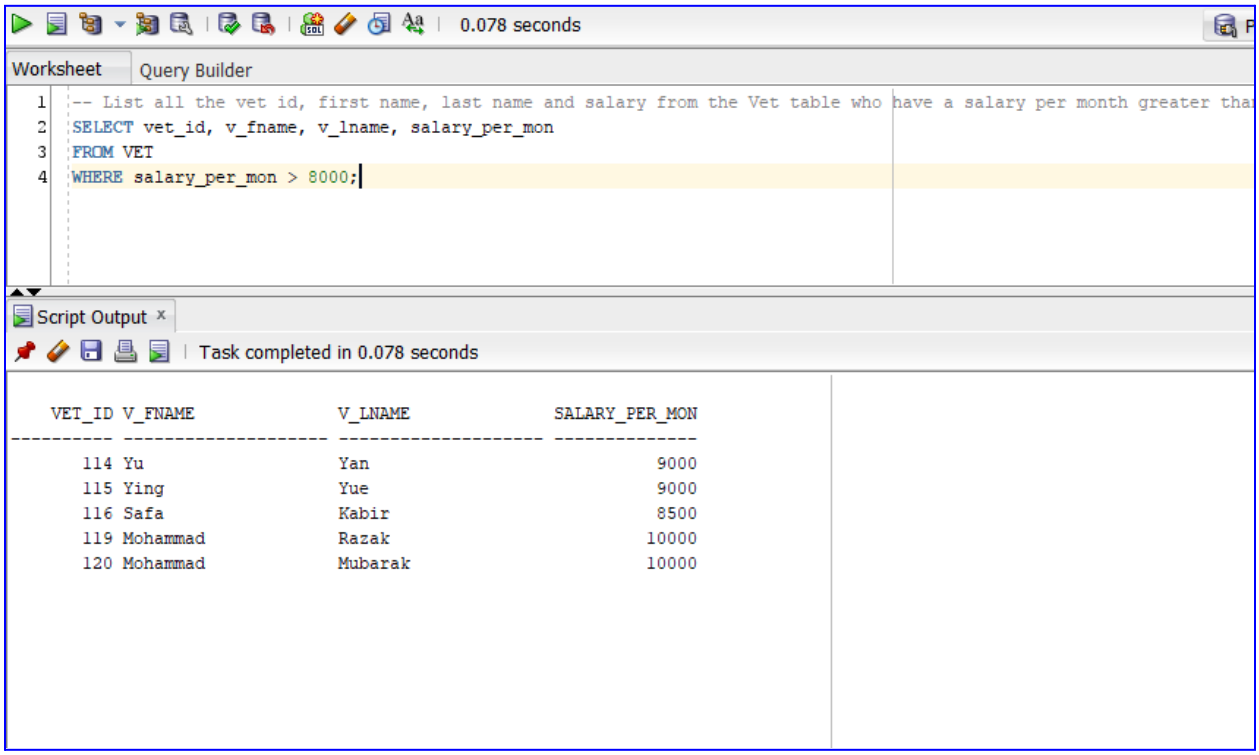
Task completed in 0.078 seconds

OWNER_ID	O_FNAME	O_LNAME	PET_ID	PET_NAME	PET_TYPE
1001	Esmee	Benitez	1101	Mini	cat
1002	Freyja	Paul	1102	Luna	cat
1002	Freyja	Paul	1103	Teddy	dog
1003	Mohammad	Iman	1104	Bill	dog
1004	Mohammad	Asraf	1105	Bella	dog
1004	Mohammad	Asraf	1106	Buddy	dog
1005	Mohammad	Arif	1108	Loki	cat
1006	Danial	Hakim	1109	Oliver	cat
1007	Raihan	Kabir	1110	Bubba	bird
1008	Mohammad	Hahmid	1107	Milo	cat
1009	Mohammad	Rakib	1111	Baldy	cat
1010	Abdur	Rahman	1112	Putih	cat

12 rows selected.

- **List all the vet id, first name, last name and salary from the *Vet* table who have a salary per month greater than *8000*.**

```
SELECT vet_id, v_fname, v_lname, salary_per_mon
FROM VET
WHERE salary_per_mon > 8000;
```



- **List all the pet's owner information from the *Owner* table who have more than one pet.**

```
SELECT *
FROM owner
WHERE owner_id IN (SELECT UNIQUE(p.owner_id)
                   FROM pet p, (SELECT owner_id, COUNT(pet_id) AS cn FROM pet GROUP
BY owner_id) s
                   WHERE p.owner_id = s.owner_id AND s.cn > 1);
```

0.095 seconds

Worksheet | Query Builder

```
-- List all the pet's owner information from the Owner table who have more than one pet.
SELECT *
FROM owner
WHERE owner_id IN (SELECT UNIQUE(p.owner_id)
                   FROM pet p, (SELECT owner_id, COUNT(pet_id) AS cn FROM pet GROUP BY owner_id) s
                   WHERE p.owner_id = s.owner_id AND s.cn > 1);
```

Script Output

Task completed in 0.095 seconds

OWNER_ID	O_FNAME	O_LNAME	O_PHONE	O_ADDRESS
1002	Freyja	Paul	01100000112	Ampang, Selangor
1004	Mohammad	Asraf	01100000114	Banting, Selangor

- **Change appointment date to 25/07/2022 where the pet's name is *'Putih'* and it is a *cat*.**

```
UPDATE appointment
SET appointment."Date" = to_date('25/07/2022', 'dd/mm/yyyy')
WHERE pet_id = (SELECT pet_id FROM pet WHERE pet_name = 'Putih' AND pet_type = 'cat');
```

0.063 seconds

Worksheet | Query Builder

```
-- Change appointment date to 25/07/2022 where the pet's name is 'Putih' and it is a cat.
UPDATE appointment
SET appointment."Date" = to_date('25/07/2022', 'dd/mm/yyyy')
WHERE pet_id = (SELECT pet_id FROM pet WHERE pet_name = 'Putih' AND pet_type = 'cat');
```

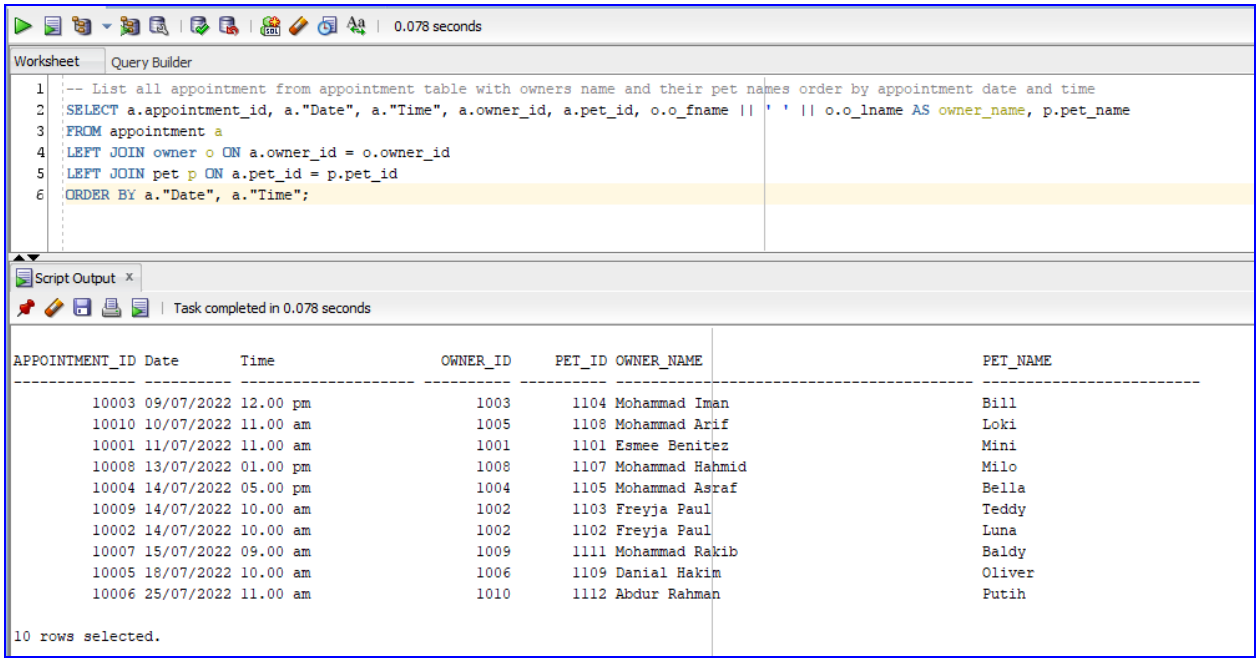
Script Output

Task completed in 0.063 seconds

```
1 row updated.
```

- **List all appointments from the appointment table with owners name and their pet names ordered by appointment date and time.**

```
SELECT a.appointment_id, a."Date", a."Time", a.owner_id, a.pet_id, o.o_fname || ' ' ||
o.o_lname AS owner_name, p.pet_name
FROM appointment a
LEFT JOIN owner o ON a.owner_id = o.owner_id
LEFT JOIN pet p ON a.pet_id = p.pet_id
ORDER BY a."Date", a."Time";
```



- **List the appointment id, date, owner name, pet name, pet type order by appointment date.**

```
SELECT a.appointment_id, a."Date", CONCAT(CONCAT(o.o_fname, ' '), o.o_lname),
p.pet_name, p.pet_type
FROM appointment a
LEFT JOIN owner o USING (owner_id)
LEFT JOIN pet p USING (pet_id);
```

0.062 seconds

Worksheet | Query Builder

```
-- List the appointment id, date, owner name, pet name, pet type order by appointment date
SELECT a.appointment_id, a."Date", CONCAT(CONCAT(o.o_fname, ' '), o.o_lname), p.pet_name, p.pet_type
FROM appointment a
LEFT JOIN owner o USING (owner_id)
LEFT JOIN pet p USING (pet_id);
```

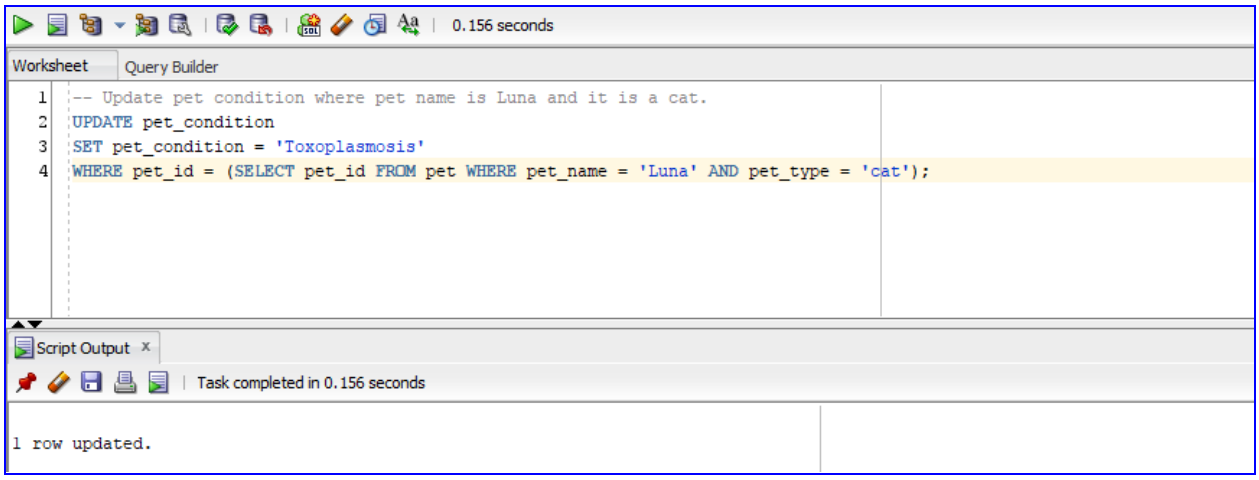
Script Output

Task completed in 0.062 seconds

```
APPOINTMENT_ID Date       CONCAT(CONCAT(O.O_FNAME,''),O.O_LNAME)    PET_NAME                       PET_TYPE
-------------- ---------- ----------------------------------------- ------------------------------ ----------
         10001 11/07/2022 Esmee Benitez                             Mini                           cat
         10002 14/07/2022 Freyja Paul                               Luna                           cat
         10009 14/07/2022 Freyja Paul                               Teddy                          dog
         10003 09/07/2022 Mohammad Iman                             Bill                           dog
         10004 14/07/2022 Mohammad Asraf                            Bella                          dog
         10008 13/07/2022 Mohammad Hahmid                           Milo                           cat
         10010 10/07/2022 Mohammad Arif                             Loki                           cat
         10005 18/07/2022 Danial Hakim                              Oliver                         cat
         10007 15/07/2022 Mohammad Rakib                            Baldy                          cat
         10006 25/07/2022 Abdur Rahman                              Putih                          cat

10 rows selected.
```

- **Update pet condition where pet name is *Luna* and it is a *cat*.**

```
UPDATE pet_condition
SET pet_condition = 'Toxoplasmosis'
WHERE pet_id = (SELECT pet_id FROM pet WHERE pet_name = 'Luna'  AND pet_type =
    'cat');
```



Procedures

- **Create a procedure which creates a new appointment. (It takes input of owner id, pet id, appointment date, appointment time, and vet id.)**

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;

CREATE OR REPLACE PROCEDURE makeAppointment
   (
      o_id IN appointment.owner_id%TYPE,
      p_id IN appointment.pet_id%TYPE,
      a_date IN appointment."Date"%TYPE,
      a_time IN appointment."Time"%TYPE,
      v_id IN appointment.vet_id %TYPE
   )
   IS
   po_id owner.owner_id%TYPE;
   is_v_exist vet.vet_id%TYPE;
   is_p_exist pet.pet_id%TYPE;
   is_o_exist owner.owner_id%TYPE;
   next_appointment_id appointment.appointment_id%TYPE;
BEGIN
   SELECT MAX(appointment_id)
   INTO next_appointment_id
   FROM appointment;
```

```
   next_appointment_id := next_appointment_id + 1;

   SELECT owner_id
   INTO po_id
   FROM pet
   WHERE pet_id = p_id;

   SELECT COUNT(*)
   INTO is_o_exist
   FROM owner
   WHERE owner_id = o_id;

   SELECT COUNT(*)
   INTO is_p_exist
   FROM pet
   WHERE pet_id = p_id;

   SELECT COUNT(*)
   INTO is_v_exist
   FROM vet
   WHERE vet_id = v_id;

   IF po_id = o_id AND is_o_exist = 1 AND is_p_exist = 1 AND is_v_exist = 1 THEN
      INSERT INTO appointment(appointment_id, owner_id, pet_id, "Date", "Time", vet_id)
      VALUES(next_appointment_id, o_id, p_id, a_date, a_time, v_id);

      DBMS_OUTPUT.PUT_LINE('New Appointment Created Successfully.');
   ELSE
      DBMS_OUTPUT.PUT_LINE('Your given Owner ID does not belong to the given Pet
ID.');
   END IF;
END;
/
```

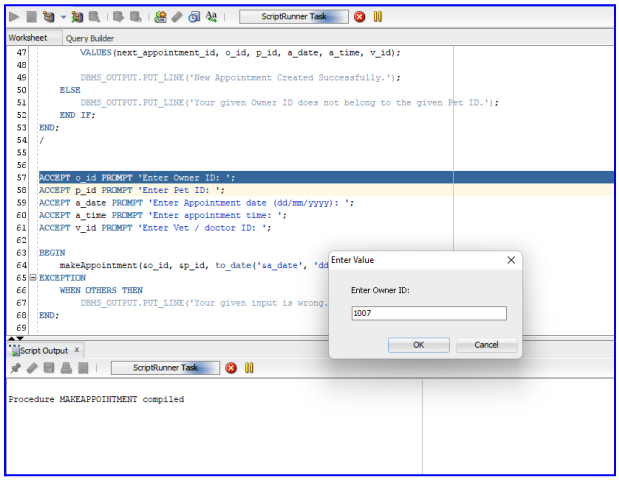
```
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT a_time PROMPT 'Enter appointment time: ';
ACCEPT v_id PROMPT 'Enter Vet / doctor ID: ';

BEGIN
   makeAppointment(&o_id, &p_id, to_date('&a_date', 'dd/mm/yyyy'), '&a_time', &v_id);
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Your input is wrong.');
END;
```

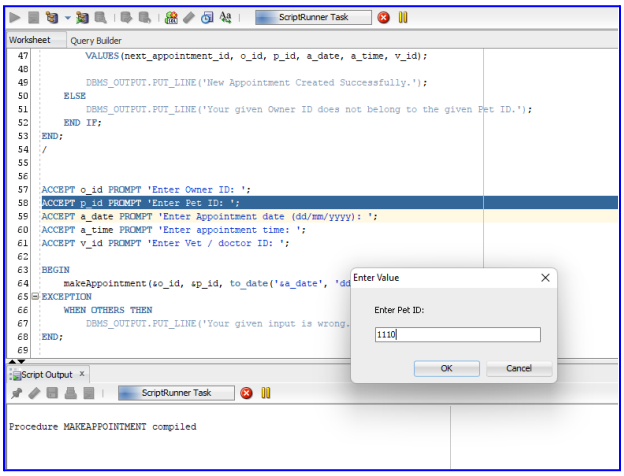
Worksheet | Query Builder

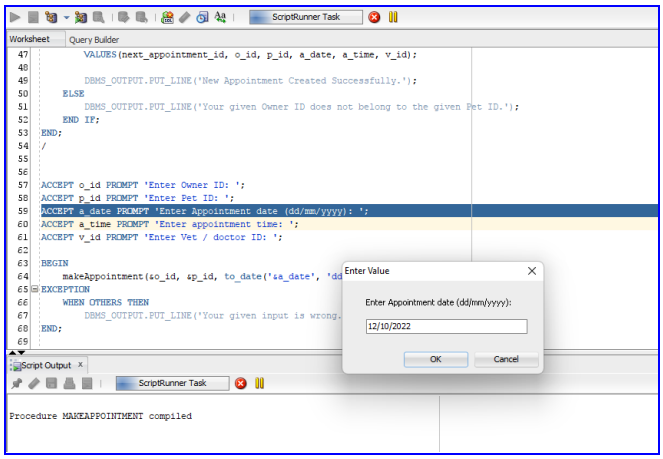
```
-- Procedure 1: Create a procedure which creates a new appointment. (It takes input of owner id, pet id, appoint
SET SERVEROUTPUT ON;
SET VERIFY OFF;
CREATE OR REPLACE PROCEDURE makeAppointment
    (
        o_id IN appointment.owner_id%TYPE,
        p_id IN appointment.pet_id%TYPE,
        a_date IN appointment."Date"%TYPE,
        a_time IN appointment."Time"%TYPE,
        v_id IN appointment.vet_id %TYPE
    )
    IS
    po_id owner.owner_id%TYPE;
    is_v_exist vet.vet_id%TYPE;
    is_p_exist pet.pet_id%TYPE;
    is_o_exist owner.owner_id%TYPE;
    next_appointment_id appointment.appointment_id%TYPE;
BEGIN
    SELECT MAX(appointment_id)
    INTO next_appointment_id
    FROM appointment;

    next_appointment_id := next_appointment_id + 1;
```

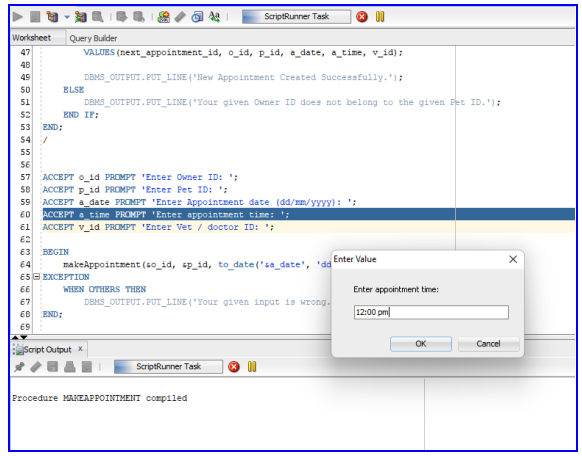
Script Output | Task completed in 0.116 seconds

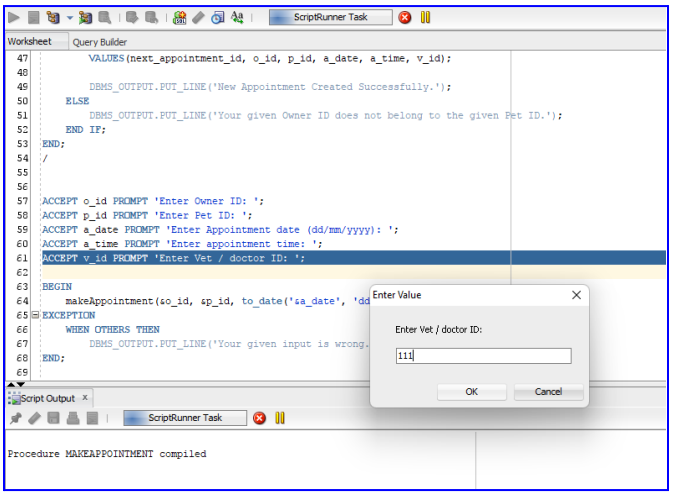
```
Procedure MAKEAPPOINTMENT compiled
```






ScriptRunner Task
Worksheet
Query Builder
VALUES(next_appointment_id, o_id, p_id, a_date, a_time, v_id);
DBMS_OUTPUT.PUT_LINE('New Appointment Created Successfully.');
ELSE
DBMS_OUTPUT.PUT_LINE('Your given Owner ID does not belong to the given Pet ID.');
END IF;
END;
/
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT a_time PROMPT 'Enter appointment time: ';
ACCEPT v_id PROMPT 'Enter Vet / doctor ID: ';
BEGIN
makeAppointment(&o_id, &p_id, to_date('&a_date', 'dd
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Your given input is wrong.
END;
Enter Value
Enter Appointment date (dd/mm/yyyy):
12/10/2022
OK
Cancel
Script Output
ScriptRunner Task
Procedure MAKEAPPOINTMENT compiled


ScriptRunner Task
Worksheet
Query Builder
VALUES(next_appointment_id, o_id, p_id, a_date, a_time, v_id);
DBMS_OUTPUT.PUT_LINE('New Appointment Created Successfully.');
ELSE
DBMS_OUTPUT.PUT_LINE('Your given Owner ID does not belong to the given Pet ID.');
END IF;
END;
/
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT a_time PROMPT 'Enter appointment time: ';
ACCEPT v_id PROMPT 'Enter Vet / doctor ID: ';
BEGIN
makeAppointment(&o_id, &p_id, to_date('&a_date', 'dd
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Your given input is wrong.
END;
Enter Value
Enter appointment time:
12:00 pm
OK
Cancel
Script Output
ScriptRunner Task
Procedure MAKEAPPOINTMENT compiled


ScriptRunner Task
Worksheet
Query Builder
VALUES(next_appointment_id, o_id, p_id, a_date, a_time, v_id);
DBMS_OUTPUT.PUT_LINE('New Appointment Created Successfully.');
ELSE
DBMS_OUTPUT.PUT_LINE('Your given Owner ID does not belong to the given Pet ID.');
END IF;
END;
/
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT a_time PROMPT 'Enter appointment time: ';
ACCEPT v_id PROMPT 'Enter Vet / doctor ID: ';
BEGIN
makeAppointment(&o_id, &p_id, to_date('&a_date', 'dd
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Your given input is wrong.
END;
Enter Value
Enter Vet / doctor ID:
111
OK
Cancel
Script Output
ScriptRunner Task
Procedure MAKEAPPOINTMENT compiled

```
Worksheet   Query Builder
47          VALUES(next_appointment_id, o_id, p_id, a_date, a_time, v_id);
48
49          DBMS_OUTPUT.PUT_LINE('New Appointment Created Successfully.');
50      ELSE
51          DBMS_OUTPUT.PUT_LINE('Your given Owner ID does not belong to the given Pet ID.');
52      END IF;
53  END;
54  /
55
56
57  ACCEPT o_id PROMPT 'Enter Owner ID: ';
58  ACCEPT p_id PROMPT 'Enter Pet ID: ';
59  ACCEPT a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
60  ACCEPT a_time PROMPT 'Enter appointment time: ';
61  ACCEPT v_id PROMPT 'Enter Vet / doctor ID: ';
62
63  BEGIN
64      makeAppointment(&o_id, &p_id, to_date('&a_date', 'dd/mm/yyyy'), '&a_time', &v_id);
65  EXCEPTION
66      WHEN OTHERS THEN
67          DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
68  END;
69
```

```
Script Output
Task completed in 112.076 seconds

Procedure MAKEAPPOINTMENT compiled

New Appointment Created Successfully.


PL/SQL procedure successfully completed.
```

- **Create a procedure that can change the appointment date. (It takes input of owner id, pet id, new appointment date, new appointment time.)**

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;

CREATE OR REPLACE PROCEDURE changeAppointment
   (
      o_id IN appointment.owner_id%TYPE,
      p_id IN appointment.pet_id%TYPE,
      u_a_date IN appointment."Date"%TYPE,
      u_a_time IN appointment."Time"%TYPE
   )
   IS
BEGIN
   UPDATE appointment
```

```
    SET "Date" = u_a_date, "Time" = u_a_time
    WHERE owner_id = o_id AND pet_id = p_id;

    DBMS_OUTPUT.PUT_LINE('Update Appointment date and time Successfully.');
END;
/
```

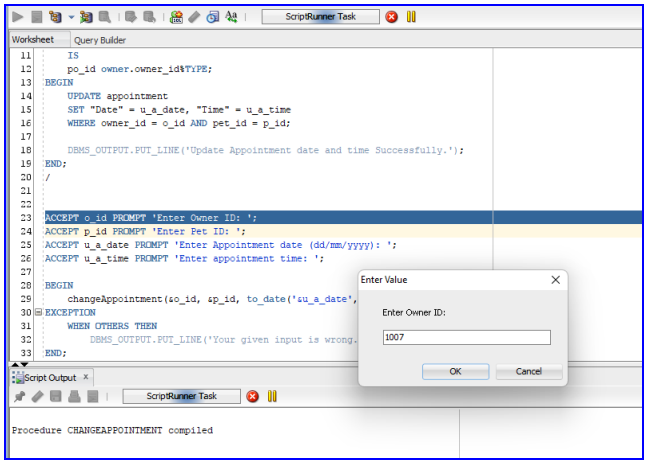
```
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT u_a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT u_a_time PROMPT 'Enter appointment time: ';

BEGIN
    changeAppointment(&o_id, &p_id, to_date('&u_a_date', 'dd/mm/yyyy'), '&u_a_time');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
END;
```

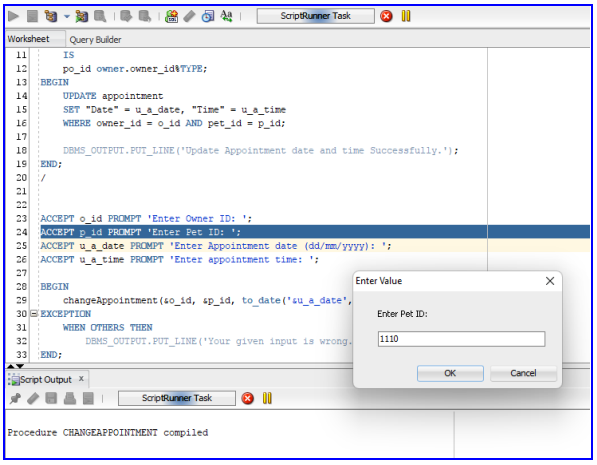
Worksheet | Query Builder

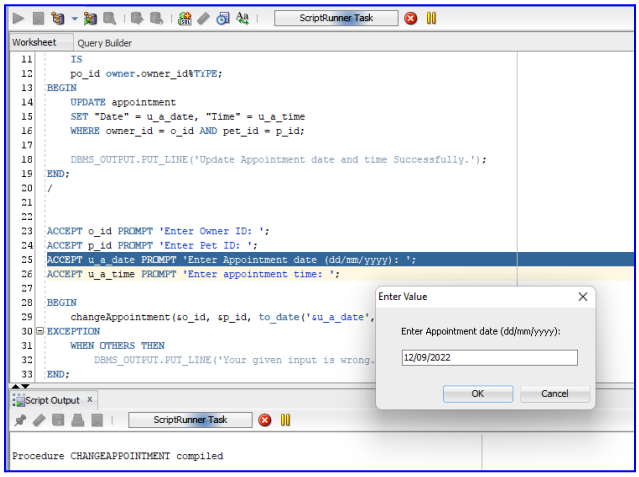
```
-- Procedure 2: Create a procedure that can change the appointment date. (It takes input of owner id, pet id, new appointment date, new appointment time.)
SET SERVEROUTPUT ON;
SET VERIFY OFF;
CREATE OR REPLACE PROCEDURE changeAppointment
    (
        o_id IN appointment.owner_id%TYPE,
        p_id IN appointment.pet_id%TYPE,
        u_a_date IN appointment."Date"%TYPE,
        u_a_time IN appointment."Time"%TYPE
    )
    IS
    po_id owner.owner_id%TYPE;
BEGIN
    UPDATE appointment
    SET "Date" = u_a_date, "Time" = u_a_time
    WHERE owner_id = o_id AND pet_id = p_id;

    DBMS_OUTPUT.PUT_LINE('Update Appointment date and time Successfully.');
END;
/


ACCEPT o_id PROMPT 'Enter Owner ID: ';
```

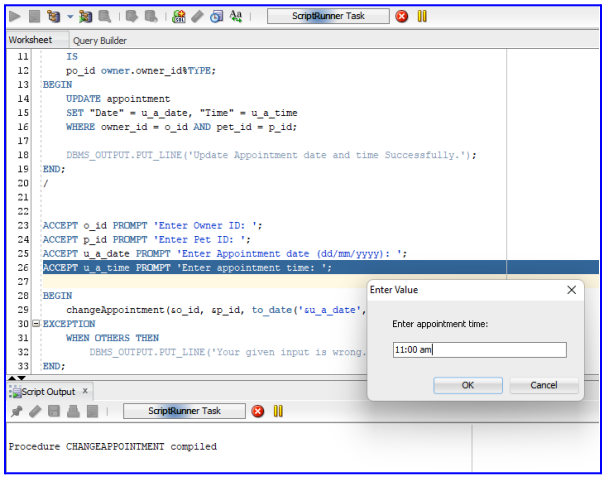
Script Output | Task completed in 0.084 seconds

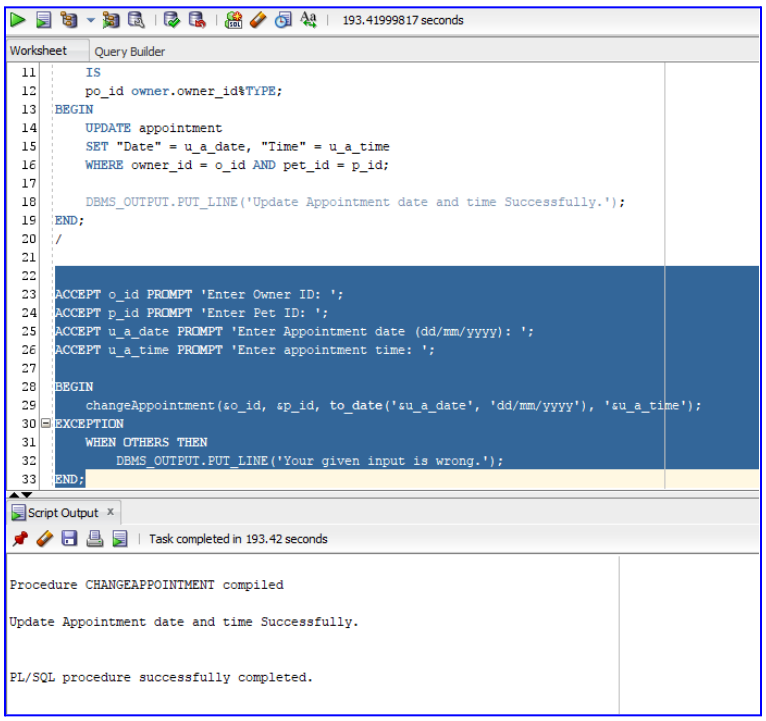
```
Procedure CHANGEAPPOINTMENT compiled
```


ScriptRunner Task
Worksheet
Query Builder
IS
po_id owner.owner_id%TYPE;
BEGIN
UPDATE appointment
SET "Date" = u_a_date, "Time" = u_a_time
WHERE owner_id = o_id AND pet_id = p_id;
DBMS_OUTPUT.PUT_LINE('Update Appointment date and time Successfully.');
END;
/
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT u_a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT u_a_time PROMPT 'Enter appointment time: ';
BEGIN
changeAppointment(&o_id, &p_id, to_date('&u_a_date',
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Your given input is wrong.
END;
Enter Value
Enter Owner ID:
1007
OK
Cancel
Script Output
ScriptRunner Task
Procedure CHANGEAPPOINTMENT compiled


ScriptRunner Task
Worksheet
Query Builder
IS
po_id owner.owner_id%TYPE;
BEGIN
UPDATE appointment
SET "Date" = u_a_date, "Time" = u_a_time
WHERE owner_id = o_id AND pet_id = p_id;
DBMS_OUTPUT.PUT_LINE('Update Appointment date and time Successfully.');
END;
/
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT u_a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT u_a_time PROMPT 'Enter appointment time: ';
BEGIN
changeAppointment(&o_id, &p_id, to_date('&u_a_date',
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Your given input is wrong.
END;
Enter Value
Enter Pet ID:
1110
OK
Cancel
Script Output
ScriptRunner Task
Procedure CHANGEAPPOINTMENT compiled


ScriptRunner Task
Worksheet
Query Builder
IS
po_id owner.owner_id%TYPE;
BEGIN
UPDATE appointment
SET "Date" = u_a_date, "Time" = u_a_time
WHERE owner_id = o_id AND pet_id = p_id;
DBMS_OUTPUT.PUT_LINE('Update Appointment date and time Successfully.');
END;
/
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT u_a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT u_a_time PROMPT 'Enter appointment time: ';
BEGIN
changeAppointment(&o_id, &p_id, to_date('&u_a_date',
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Your given input is wrong.
END;
Enter Value
Enter Appointment date (dd/mm/yyyy):
12/09/2022
OK
Cancel
Script Output
ScriptRunner Task
Procedure CHANGEAPPOINTMENT compiled


ScriptRunner Task
Worksheet
Query Builder
IS
po_id owner.owner_id%TYPE;
BEGIN
UPDATE appointment
SET "Date" = u_a_date, "Time" = u_a_time
WHERE owner_id = o_id AND pet_id = p_id;
DBMS_OUTPUT.PUT_LINE('Update Appointment date and time Successfully.');
END;
/
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT u_a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT u_a_time PROMPT 'Enter appointment time: ';
BEGIN
changeAppointment(&o_id, &p_id, to_date('&u_a_date',
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Your given input is wrong.
END;
Enter Value
Enter appointment time:
11:00 am
OK
Cancel
Script Output
ScriptRunner Task
Procedure CHANGEAPPOINTMENT compiled


193.41999817 seconds
Worksheet
Query Builder
IS
po_id owner.owner_id%TYPE;
BEGIN
UPDATE appointment
SET "Date" = u_a_date, "Time" = u_a_time
WHERE owner_id = o_id AND pet_id = p_id;
DBMS_OUTPUT.PUT_LINE('Update Appointment date and time Successfully.');
END;
/
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';
ACCEPT u_a_date PROMPT 'Enter Appointment date (dd/mm/yyyy): ';
ACCEPT u_a_time PROMPT 'Enter appointment time: ';
BEGIN
changeAppointment(&o_id, &p_id, to_date('&u_a_date', 'dd/mm/yyyy'), '&u_a_time');
EXCEPTION
WHEN OTHERS THEN
DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
END;
Script Output
Task completed in 193.42 seconds
Procedure CHANGEAPPOINTMENT compiled
Update Appointment date and time Successfully.
PL/SQL procedure successfully completed.

- **Delete the appointment. (It takes input of owner id, pet id.)**

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;

CREATE OR REPLACE PROCEDURE deleteAppointment
   (
      o_id IN appointment.owner_id%TYPE,
      p_id IN appointment.pet_id%TYPE
   )
   IS
BEGIN
   DELETE FROM appointment
   WHERE owner_id = o_id AND pet_id = p_id;

   DBMS_OUTPUT.PUT_LINE('Delete Appointment Successfully.');
END;
/
```

```
ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';

BEGIN
   deleteAppointment(&o_id, &p_id);
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
END;
```

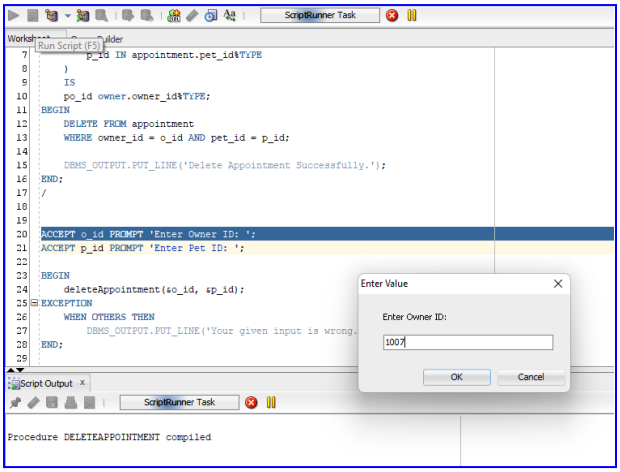
0.163 seconds

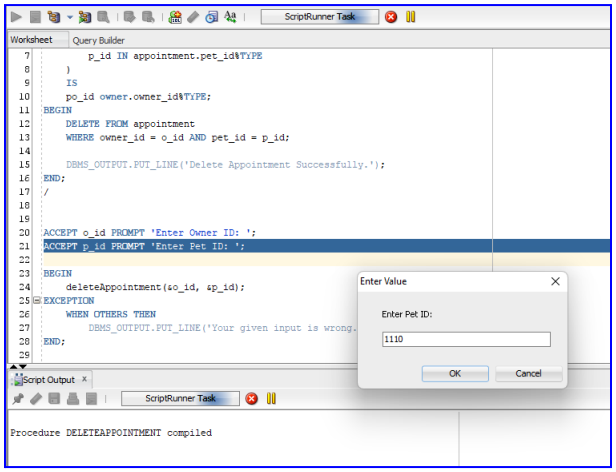
Worksheet | Query Builder

```
-- Procedure 3: Delete the appointment. (It takes input of owner id, pet id.)
SET SERVEROUTPUT ON;
SET VERIFY OFF;
CREATE OR REPLACE PROCEDURE deleteAppointment
    (
        o_id IN appointment.owner_id%TYPE,
        p_id IN appointment.pet_id%TYPE
    )
    IS
    po_id owner.owner_id%TYPE;
BEGIN
    DELETE FROM appointment
    WHERE owner_id = o_id AND pet_id = p_id;

    DBMS_OUTPUT.PUT_LINE('Delete Appointment Successfully.');
END;
/


ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';

BEGIN
```

Script Output | Task completed in 0.163 seconds

```
Procedure DELETEAPPOINTMENT compiled
```





76.41500092 seconds

Worksheet | Query Builder

```
        p_id IN appointment.pet_id%TYPE
    )
    IS
    po_id owner.owner_id%TYPE;
BEGIN
    DELETE FROM appointment
    WHERE owner_id = o_id AND pet_id = p_id;

    DBMS_OUTPUT.PUT_LINE('Delete Appointment Successfully.');
END;
/


ACCEPT o_id PROMPT 'Enter Owner ID: ';
ACCEPT p_id PROMPT 'Enter Pet ID: ';

BEGIN
    deleteAppointment(&o_id, &p_id);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
END;
```

Script Output | Task completed in 76.415 seconds

```
Procedure DELETEAPPOINTMENT compiled

Delete Appointment Successfully.


PL/SQL procedure successfully completed.
```

- **Add a new owner. (It takes input of the owner's first name, last name, phone number, and address.)**

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;

CREATE OR REPLACE PROCEDURE newOwner
   (
      ow_fname IN owner.o_fname%TYPE,
      ow_lname IN owner.o_lname%TYPE,
      ow_phone IN owner.o_phone%TYPE,
      ow_address IN owner.o_address%TYPE
```

```
    )
    IS
    next_ow_id owner.owner_id%TYPE;
BEGIN
    SELECT MAX(owner_id)
    INTO next_ow_id
    FROM owner;

    next_ow_id := next_ow_id + 1;

    INSERT INTO owner(owner_id, o_fname, o_lname, o_phone, o_address)
    VALUES(next_ow_id, ow_fname, ow_lname, ow_phone, ow_address);

    DBMS_OUTPUT.PUT_LINE('New owner created Successfully.');
END;
/
```

```
ACCEPT ow_fname PROMPT 'Enter your first name: ';
ACCEPT ow_lname PROMPT 'Enter your last name: ';
ACCEPT ow_phone PROMPT 'Enter your phone number: ';
ACCEPT ow_address PROMPT 'Enter your address: ';

BEGIN
    newOwner('&ow_fname', '&ow_lname', '&ow_phone', '&ow_address');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
END;
```

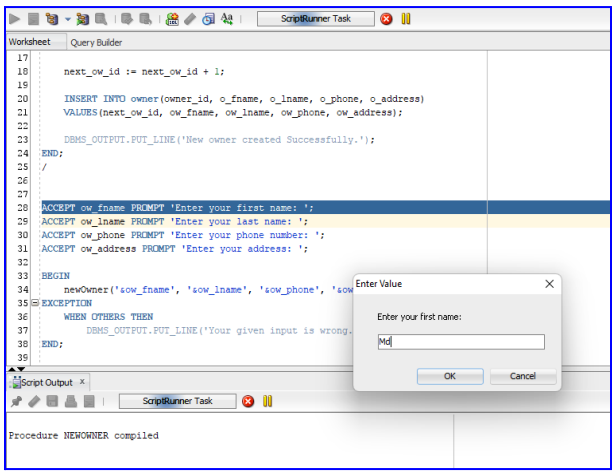
0.118 seconds

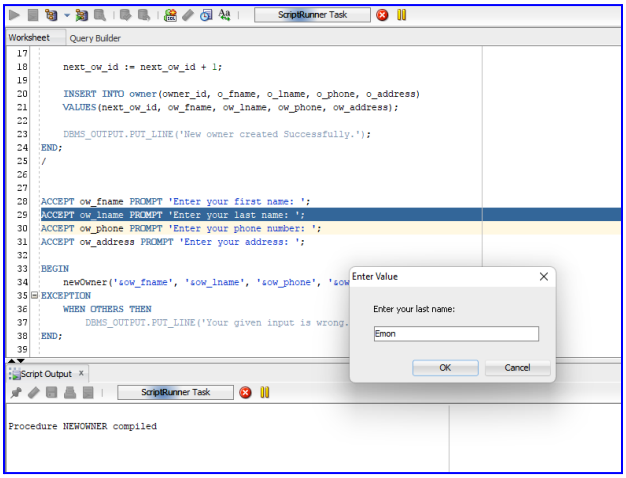
Worksheet | Query Builder

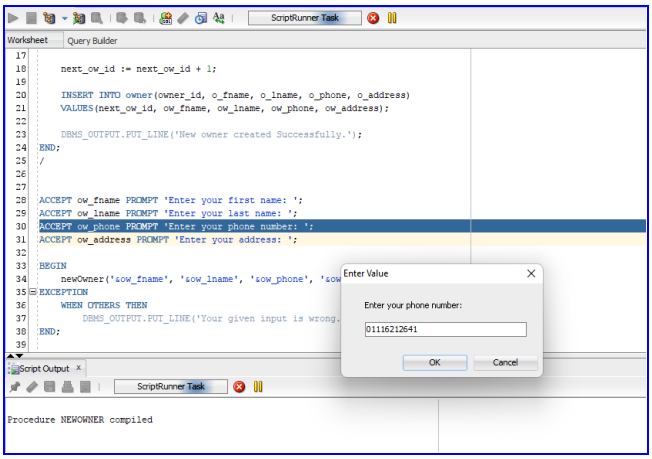
```
-- Procedure 4: Add a new owner. (It takes input of the owner's first name, last name, phone number, and address.)
SET SERVEROUTPUT ON;
SET VERIFY OFF;
CREATE OR REPLACE PROCEDURE newOwner
    (
        ow_fname IN owner.o_fname%TYPE,
        ow_lname IN owner.o_lname%TYPE,
        ow_phone IN owner.o_phone%TYPE,
        ow_address IN owner.o_address%TYPE
    )
    IS
    next_ow_id owner.owner_id%TYPE;
BEGIN
    SELECT MAX(owner_id)
    INTO next_ow_id
    FROM owner;

    next_ow_id := next_ow_id + 1;

    INSERT INTO owner(owner_id, o_fname, o_lname, o_phone, o_address)
    VALUES(next_ow_id, ow_fname, ow_lname, ow_phone, ow_address);

    DBMS_OUTPUT.PUT_LINE('New owner created Successfully.');
```

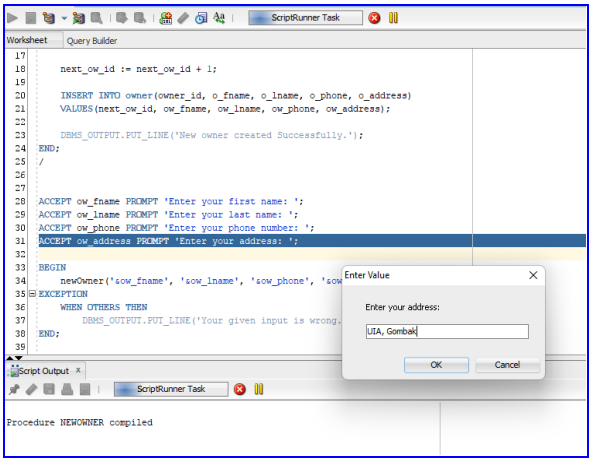
Script Output | Task completed in 0.118 seconds

```
Procedure NEWOWNER compiled
```









181.14399719 seconds

Worksheet | Query Builder

```
    next_ow_id := next_ow_id + 1;

    INSERT INTO owner(owner_id, o_fname, o_lname, o_phone, o_address)
    VALUES(next_ow_id, ow_fname, ow_lname, ow_phone, ow_address);

    DBMS_OUTPUT.PUT_LINE('New owner created Successfully.');
END;
/


ACCEPT ow_fname PROMPT 'Enter your first name: ';
ACCEPT ow_lname PROMPT 'Enter your last name: ';
ACCEPT ow_phone PROMPT 'Enter your phone number: ';
ACCEPT ow_address PROMPT 'Enter your address: ';

BEGIN
    newOwner('&ow_fname', '&ow_lname', '&ow_phone', '&ow_address');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
END;
```

Script Output

Task completed in 181.144 seconds

```
Procedure NEWOWNER compiled

New owner created Successfully.


PL/SQL procedure successfully completed.
```

- **Delete an owner. (It takes input of the owner's id.)**

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;

CREATE OR REPLACE PROCEDURE deleteOwner
   (
      ow_id IN owner.owner_id%TYPE
   )
   IS
BEGIN
   DELETE FROM owner
   WHERE owner_id = ow_id;

   DELETE FROM pet
   WHERE owner_id = ow_id;

   DBMS_OUTPUT.PUT_LINE('Delete owner Successfully.');
END;
/
```

```
ACCEPT ow_id PROMPT 'Enter Owner ID: ';

BEGIN
   deleteOwner(&ow_id);
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
END;
```

0.106 seconds

Worksheet | Query Builder

```
-- Procedure 5: Delete an owner. (It takes input of the owner's id.)
SET SERVEROUTPUT ON;
SET VERIFY OFF;
CREATE OR REPLACE PROCEDURE deleteOwner
    (
        ow_id IN owner.owner_id%TYPE
    )
    IS
BEGIN
    DELETE FROM owner
    WHERE owner_id = ow_id;

    DELETE FROM pet
    WHERE owner_id = ow_id;

    DBMS_OUTPUT.PUT_LINE('Delete owner created Successfully.');
END;
/


ACCEPT ow_id PROMPT 'Enter Owner ID: ';

BEGIN
```

Script Output

Task completed in 0.106 seconds

```
Procedure DELETEOWNER compiled
```

Project~2 | OnlineVetAppSys_procedure.sql

ScriptRunner Task

Worksheet | Query Builder

```
        ow_id IN owner.owner_id%TYPE
    )
    IS
BEGIN
    DELETE FROM owner
    WHERE owner_id = ow_id;

    DELETE FROM pet
    WHERE owner_id = ow_id;

    DBMS_OUTPUT.PUT_LINE('Delete owner created Successfully.');
END;
/


ACCEPT ow_id PROMPT 'Enter Owner ID: ';

BEGIN
    deleteOwner(&ow_id);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Your given input is wrong.
END;
```

Enter Value

Enter Owner ID:

1012

OK | Cancel

Script Output

ScriptRunner Task

```
Procedure DELETEOWNER compiled
```

```
        ow_id IN owner.owner_id%TYPE
    )
    IS
BEGIN
    DELETE FROM owner
    WHERE owner_id = ow_id;

    DELETE FROM pet
    WHERE owner_id = ow_id;

    DBMS_OUTPUT.PUT_LINE('Delete owner created Successfully.');
END;
/


ACCEPT ow_id PROMPT 'Enter Owner ID: ';

BEGIN
    deleteOwner(&ow_id);
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Your given input is wrong.');
END;
```

Script Output

Task completed in 34.67 seconds

```
Procedure DELETEOWNER compiled

Delete owner created Successfully.


PL/SQL procedure successfully completed.
```

Functions

- **Create a function to give a discount on the charge of an appointment. Give 20% discount to those who live in Selangor, 10% discount to those who live in Kuala Lumpur.**

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;

CREATE OR REPLACE FUNCTION getDiscount (ow_address IN
owner.o_address%TYPE)
   RETURN NUMBER
   IS
```

```
   discount NUMBER;
BEGIN
   IF ow_address LIKE '%Selangor%' THEN discount := 0.2;
      ELSIF ow_address LIKE '%Kuala Lumpur%' THEN discount := 0.2;
      ELSE discount := 0;
   END IF;
   RETURN discount;
END;
/
```

```
ACCEPT ow_id PROMPT 'Enter Owner ID: ';
DECLARE
   o_id owner.owner_id%TYPE;
   ow_fname owner.o_fname%TYPE;
   ow_lname owner.o_lname%TYPE;
   ow_address owner.o_address%TYPE;
   charge_of_vet vet.charge_per_app%TYPE;
   discount NUMBER;
   discount_bill NUMBER;
BEGIN
   o_id := &ow_id;

   SELECT charge_per_app
   INTO charge_of_vet
   FROM vet
   WHERE vet_id = (SELECT vet_id FROM appointment WHERE owner_id = o_id);

   SELECT o_fname, o_lname, o_address
   INTO ow_fname, ow_lname, ow_address
   FROM owner
   WHERE owner_id = o_id;

   discount := getDiscount(ow_address);
   discount_bill := charge_of_vet - (charge_of_vet * discount);

   DBMS_OUTPUT.PUT_LINE('Pet Owner Name: ' || ow_fname || ' ' || ow_lname);
   DBMS_OUTPUT.PUT_LINE('Pet Owner Address: ' || ow_address);
   DBMS_OUTPUT.PUT_LINE('Vet fee: ' || charge_of_vet);
   DBMS_OUTPUT.PUT_LINE('Eligible Discount: ' || (discount * 100) || '%' || chr(10));

   DBMS_OUTPUT.PUT_LINE('Vet fee after discount: ' || discount_bill);
EXCEPTION
   WHEN OTHERS THEN
      DBMS_OUTPUT.PUT_LINE('Your given Owner ID does not have any appointment.
Or Your given Owner ID is Invalid.');
END;
```

0.125 seconds

Worksheet | Query Builder

```oracle_plsql
-- Function 1: Create a function to give a discount on the charge of an appointment. Give 20% discount to those who live in Selangor,
SET SERVEROUTPUT ON;
SET VERIFY OFF;
CREATE OR REPLACE FUNCTION getDiscount (ow_address IN owner.o_address%TYPE)
    RETURN NUMBER
    IS
    discount NUMBER;
BEGIN
    IF ow_address LIKE '%Selangor%' THEN discount := 0.2;
        ELSIF ow_address LIKE '%Kuala Lumpur%' THEN discount := 0.2;
        ELSE discount := 0;
    END IF;
    RETURN discount;
END;


ACCEPT ow_id PROMPT 'Enter Owner ID: ';
DECLARE
    o_id owner.owner_id%TYPE;
    ow_fname owner.o_fname%TYPE;
    ow_lname owner.o_lname%TYPE;
    ow_address owner.o_address%TYPE;
    charge_of_vet vet.charge_per_app%TYPE;
```

Script Output

Task completed in 0.125 seconds

```
Function GETDISCOUNT compiled
```

OnlineVetAppSys_function.sql | Project~3 | Assignment_5.sql

ScriptRunner Task

Worksheet | Query Builder

```
    discount NUMBER;
BEGIN
    IF ow_address LIKE '%Selangor%' THEN discount := 0.2;
        ELSIF ow_address LIKE '%Kuala Lumpur%' THEN discount := 0.2;
        ELSE discount := 0;
    END IF;
    RETURN discount;
END;


ACCEPT ow_id PROMPT 'Enter Owner ID: ';
DECLARE
    o_id owner.owner_id%TYPE;
    ow_fname owner.o_fname%TYPE;
    ow_lname owner.o_lname%TYPE;
    ow_address owner.o_address%TYPE;
    charge_of_vet vet.charge_per_app%TYPE;
    discount NUMBER;
    discount_bill NUMBER;
BEGIN
    o_id := &ow_id;

    SELECT charge_per_app
```

Enter Value

Enter Owner ID:

1001

OK | Cancel

Script Output

ScriptRunner Task

```
Function GETDISCOUNT compiled
```

26.43499947 seconds

Worksheet | Query Builder

```
BEGIN
    o_id := &ow_id;

    SELECT charge_per_app
    INTO charge_of_vet
    FROM vet
    WHERE vet_id = (SELECT vet_id FROM appointment WHERE owner_id = o_id);

    SELECT o_fname, o_lname, o_address
    INTO ow_fname, ow_lname, ow_address
    FROM owner
    WHERE owner_id = o_id;

    discount := getDiscount(ow_address);
    discount_bill := charge_of_vet - (charge_of_vet * discount);

    DBMS_OUTPUT.PUT_LINE('Pet Owner Name: ' || ow_fname || ' ' || ow_lname);
    DBMS_OUTPUT.PUT_LINE('Pet Owner Address: ' || ow_address);
    DBMS_OUTPUT.PUT_LINE('Vet fee: ' || charge_of_vet);
    DBMS_OUTPUT.PUT_LINE('Eligible Discount: ' || (discount * 100) || '%' || chr(10));

    DBMS_OUTPUT.PUT_LINE('Vet fee after discount: ' || discount_bill);
EXCEPTION
```

Script Output | Task completed in 26.435 seconds

```
Function GETDISCOUNT compiled

Pet Owner Name: Esmee Benitez
Pet Owner Address: Gombak, Selangor
Vet fee: 80
Eligible Discount: 20%

Vet fee after discount: 64


PL/SQL procedure successfully completed.
```

- Create a function to check how many appointments a vet in a month. Function takes vet_id, month name, and year as input.

```
SET SERVEROUTPUT ON;
SET VERIFY OFF;

CREATE OR REPLACE FUNCTION countAppiontmentOfVet
    (
        v_id IN vet.vet_id%TYPE,
        month_name IN VARCHAR2,
        g_year IN VARCHAR2
    )
    RETURN NUMBER
    IS
    t_count NUMBER;
    s_date VARCHAR2(30);
    f_date VARCHAR2(30);
```

```
BEGIN
   IF month_name = 'January' THEN s_date := '01/01/' || g_year; f_date := '31/01/' ||
g_year;
       ELSIF month_name = 'February' THEN s_date := '01/02/' || g_year; f_date := '28/02/'
|| g_year;
       ELSIF month_name = 'March' THEN s_date := '01/03/' || g_year; f_date := '31/03/' ||
g_year;
       ELSIF month_name = 'April' THEN s_date := '01/04/' || g_year; f_date := '30/04/' ||
g_year;
       ELSIF month_name = 'May' THEN s_date := '01/05/' || g_year; f_date := '31/05/' ||
g_year;
       ELSIF month_name = 'June' THEN s_date := '01/06/' || g_year; f_date := '30/06/' ||
g_year;
       ELSIF month_name = 'July' THEN s_date := '01/07/' || g_year; f_date := '31/07/' ||
g_year;
       ELSIF month_name = 'August' THEN s_date := '01/08/' || g_year; f_date := '31/08/' ||
g_year;
       ELSIF month_name = 'September' THEN s_date := '01/09/' || g_year; f_date :=
'30/09/' || g_year;
       ELSIF month_name = 'October' THEN s_date := '01/10/' || g_year; f_date := '31/10/' ||
g_year;
       ELSIF month_name = 'November' THEN s_date := '01/11/' || g_year; f_date := '30/11/'
|| g_year;
       ELSIF month_name = 'December' THEN s_date := '01/12/' || g_year; f_date :=
'31/12/' || g_year;
       ELSE DBMS_OUTPUT.PUT_LINE('You Enter a Invalid month name.');
   END IF;

   SELECT COUNT(*)
   INTO t_count
   FROM appointment
   WHERE vet_id = v_id
   AND "Date" BETWEEN to_date(s_date, 'dd/mm/yyyy') AND to_date(f_date,
'dd/mm/yyyy');

   RETURN t_count;
END;
/
```

```
ACCEPT v_id PROMPT 'Enter a Vet ID: ';
ACCEPT month_name PROMPT 'Enter a month (eg. January, February ): ';
ACCEPT g_year PROMPT 'Enter the year: ';
DECLARE
   vid vet.vet_id%TYPE;
   vet_fname vet.v_fname%TYPE;
   vet_lname vet.v_lname%TYPE;
```

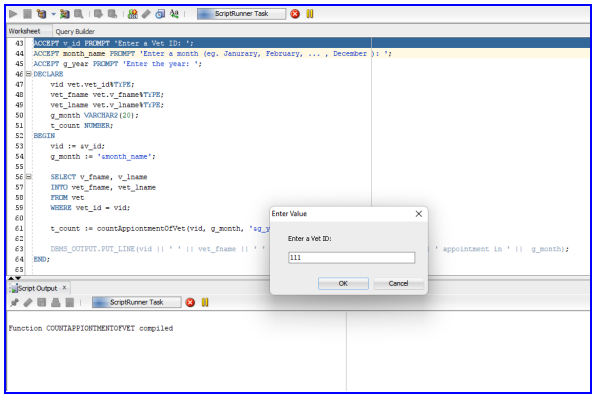
```
    g_month VARCHAR2(20);
    t_count NUMBER;
BEGIN
    vid := &v_id;
    g_month := '&month_name';

    SELECT v_fname, v_lname
    INTO vet_fname, vet_lname
    FROM vet
    WHERE vet_id = vid;

    t_count := countAppiontmentOfVet(vid, g_month, '&g_year');

    DBMS_OUTPUT.PUT_LINE(vid || ' ' || vet_fname || ' ' || vet_lname || ' have ' || t_count || '
appointment in ' ||  g_month);
END;
```

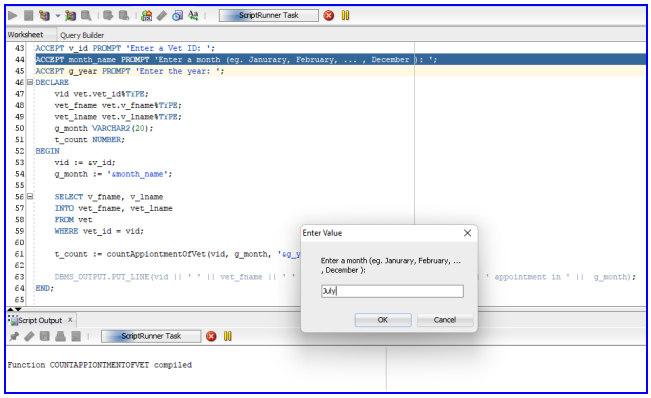
Worksheet | Query Builder

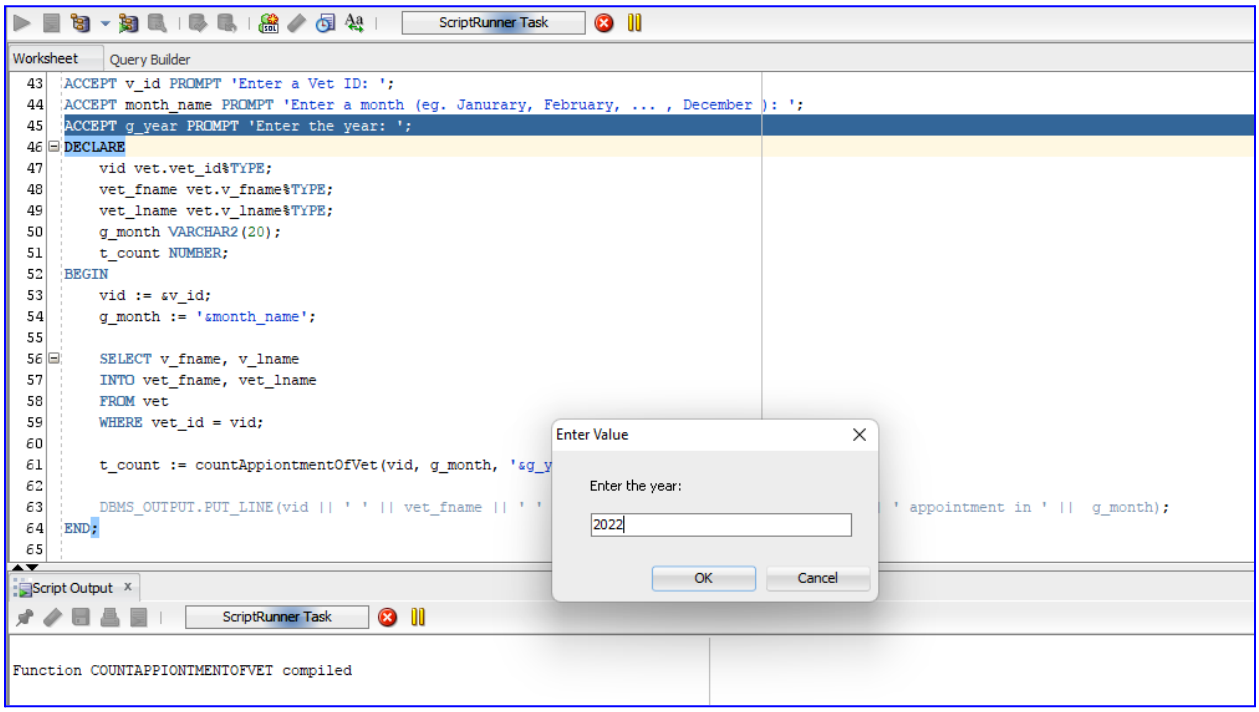
```
-- Function - 2: Create a function to check how many appointments have each vet. Function only take vet_id as input.
SET SERVEROUTPUT ON;
SET VERIFY OFF;
CREATE OR REPLACE FUNCTION countAppiontmentOfVet
    (
        v_id IN vet.vet_id%TYPE,
        month_name IN VARCHAR2,
        g_year IN VARCHAR2
    )
    RETURN NUMBER
    IS
    t_count NUMBER;
    s_date VARCHAR2(30);
    f_date VARCHAR2(30);

BEGIN
    IF month_name = 'January' THEN s_date := '01/01/' || g_year; f_date := '31/01/' || g_year;
        ELSIF month_name = 'February' THEN s_date := '01/02/' || g_year; f_date := '28/02/' || g_year;
        ELSIF month_name = 'March' THEN s_date := '01/03/' || g_year; f_date := '31/03/' || g_year;
        ELSIF month_name = 'April' THEN s_date := '01/04/' || g_year; f_date := '30/04/' || g_year;
        ELSIF month_name = 'May' THEN s_date := '01/05/' || g_year; f_date := '31/05/' || g_year;
        ELSIF month_name = 'June' THEN s_date := '01/06/' || g_year; f_date := '30/06/' || g_year;
        ELSIF month_name = 'July' THEN s_date := '01/07/' || g_year; f_date := '31/07/' || g_year;
```

Script Output | Task completed in 0.11 seconds

```
Function COUNTAPPIONTMENTOFVET compiled
```


ScriptRunner Task
Worksheet
Query Builder
ACCEPT v_id PROMPT 'Enter a Vet ID: ';
ACCEPT month_name PROMPT 'Enter a month (eg. Janurary, February, ... , December): ';
ACCEPT g_year PROMPT 'Enter the year: ';
DECLARE
vid vet.vet_id%TYPE;
vet_fname vet.v_fname%TYPE;
vet_lname vet.v_lname%TYPE;
g_month VARCHAR2(20);
t_count NUMBER;
BEGIN
vid := &v_id;
g_month := '&month_name';
SELECT v_fname, v_lname
INTO vet_fname, vet_lname
FROM vet
WHERE vet_id = vid;
t_count := countAppiontmentOfVet(vid, g_month, '&g_y
DBMS_OUTPUT.PUT_LINE(vid || ' ' || vet_fname || ' '
' appointment in ' || g_month);
END;
Enter Value
Enter a Vet ID:
111
OK
Cancel
Script Output
ScriptRunner Task
Function COUNTAPPIONTMENTOFVET compiled


ScriptRunner Task
Worksheet
Query Builder
ACCEPT v_id PROMPT 'Enter a Vet ID: ';
ACCEPT month_name PROMPT 'Enter a month (eg. Janurary, February, ... , December): ';
ACCEPT g_year PROMPT 'Enter the year: ';
DECLARE
vid vet.vet_id%TYPE;
vet_fname vet.v_fname%TYPE;
vet_lname vet.v_lname%TYPE;
g_month VARCHAR2(20);
t_count NUMBER;
BEGIN
vid := &v_id;
g_month := '&month_name';
SELECT v_fname, v_lname
INTO vet_fname, vet_lname
FROM vet
WHERE vet_id = vid;
t_count := countAppiontmentOfVet(vid, g_month, '&g_y
DBMS_OUTPUT.PUT_LINE(vid || ' ' || vet_fname || ' '
' appointment in ' || g_month);
END;
Enter Value
Enter a month (eg. Janurary, February, ... , December):
July
OK
Cancel
Script Output
ScriptRunner Task
Function COUNTAPPIONTMENTOFVET compiled


ScriptRunner Task
Worksheet
Query Builder
43 ACCEPT v_id PROMPT 'Enter a Vet ID: ';
44 ACCEPT month_name PROMPT 'Enter a month (eg. Janurary, February, ... , December): ';
45 ACCEPT g_year PROMPT 'Enter the year: ';
46 DECLARE
47 vid vet.vet_id%TYPE;
48 vet_fname vet.v_fname%TYPE;
49 vet_lname vet.v_lname%TYPE;
50 g_month VARCHAR2(20);
51 t_count NUMBER;
52 BEGIN
53 vid := &v_id;
54 g_month := '&month_name';
55
56 SELECT v_fname, v_lname
57 INTO vet_fname, vet_lname
58 FROM vet
59 WHERE vet_id = vid;
60
61 t_count := countAppiontmentOfVet(vid, g_month, '&g_y
62
63 DBMS_OUTPUT.PUT_LINE(vid || ' ' || vet_fname || ' '
' appointment in ' || g_month);
64 END;
65
Enter Value
Enter the year:
2022
OK
Cancel
Script Output
ScriptRunner Task
Function COUNTAPPIONTMENTOFVET compiled

61.26900101 seconds

Worksheet | Query Builder

```
/


ACCEPT v_id PROMPT 'Enter a Vet ID: ';
ACCEPT month_name PROMPT 'Enter a month (eg. Janurary, February, ... , December ): ';
ACCEPT g_year PROMPT 'Enter the year: ';
DECLARE
    vid vet.vet_id%TYPE;
    vet_fname vet.v_fname%TYPE;
    vet_lname vet.v_lname%TYPE;
    g_month VARCHAR2(20);
    t_count NUMBER;
BEGIN
    vid := &v_id;
    g_month := '&month_name';

    SELECT v_fname, v_lname
    INTO vet_fname, vet_lname
    FROM vet
    WHERE vet_id = vid;

    t_count := countAppiontmentOfVet(vid, g_month, '&g_year');

```

Script Output

Task completed in 61.269 seconds

```
Function COUNTAPPIONTMENTOFVET compiled

111 Tom Harij have 3 appointment in July


PL/SQL procedure successfully completed.
```