ANIMAL SHELTER DATABASE SYSTEM PROJECT - PART 5

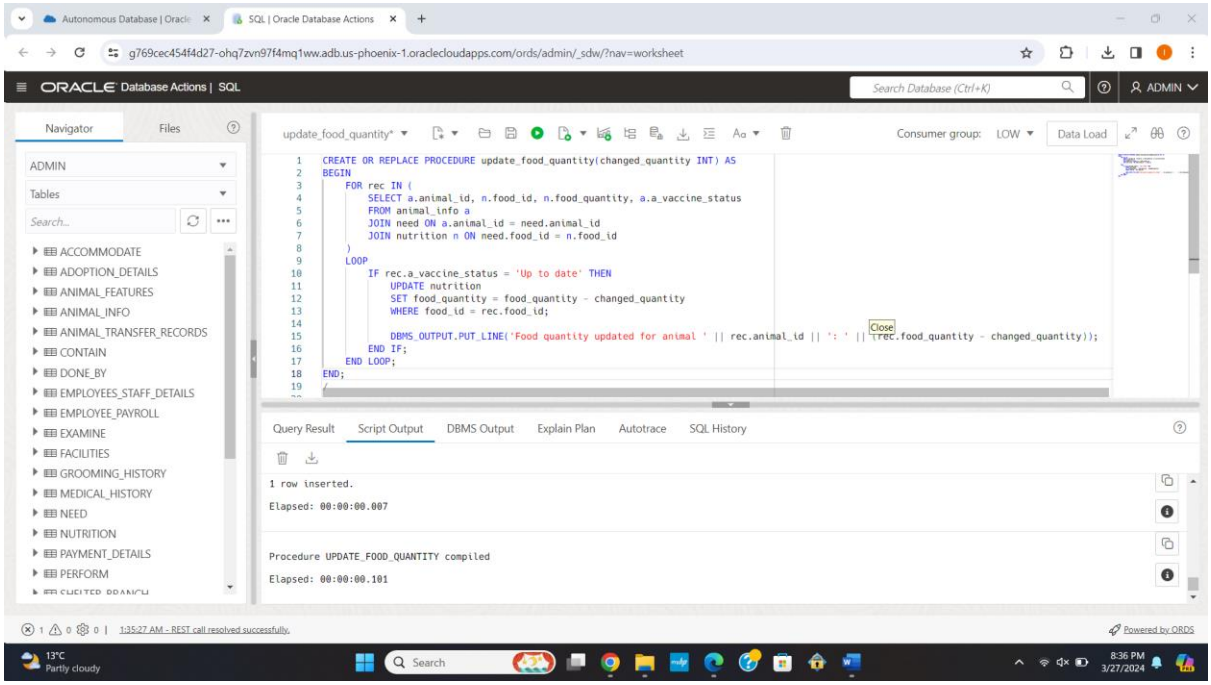
Updated Assumptions:

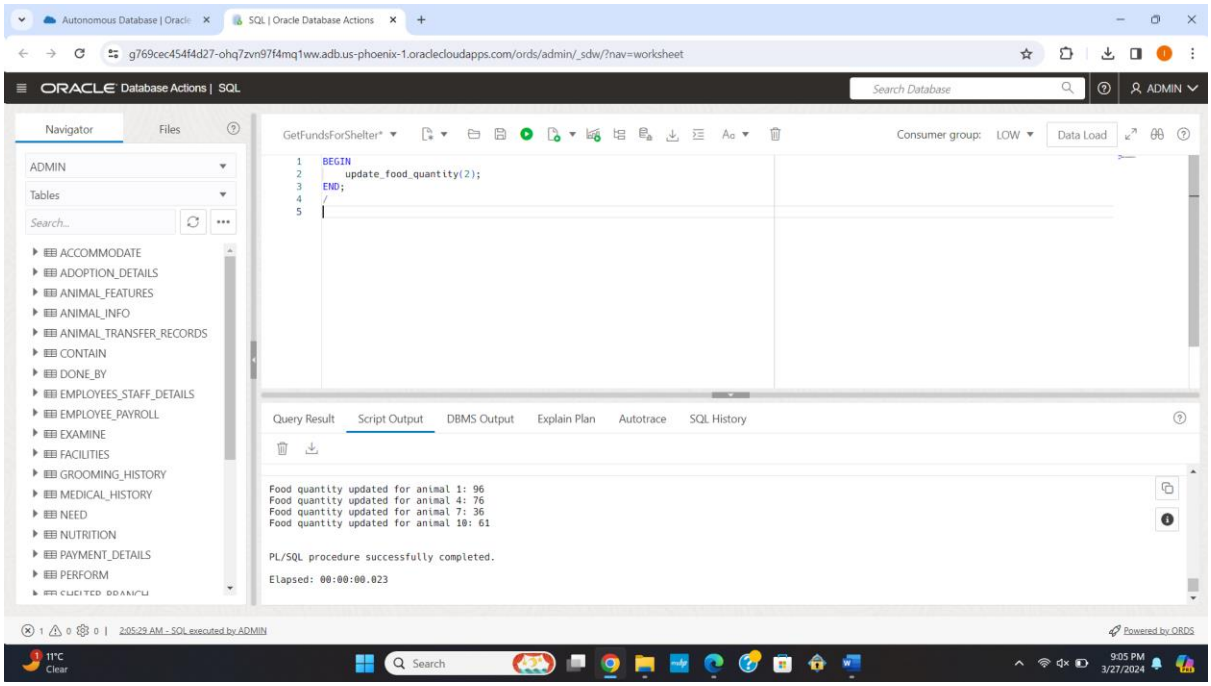
1. Animals can be transferred from one branch to another
2. Each shelter branch will have multiple employees
3. Each shelter branch will have a branch manager
4. Employees work only in one shelter branch only (no transfer of employees)
5. An animal can be adopted/surrendered to the shelter for various reasons multiple times
6. A vaccination is given each time an animal is detected with any disease
7. Sponsors can support multiple shelter branches
8. Full payment must be done in a single payment for adoption of an animal
9. Each medical condition of an animal is treated by only one veterinarian
10. Veterinarians can treat multiple conditions of multiple animals
11. Each transfer of an animal involves transfer from one branch to another
12. Each veterinary doctor can have multiple specializations
13. All contacts should not contain spaces
14. Employees work on hourly basis and hourly pay varies from employee to employee
15. Any animal is considered as ready for adoption if its vaccination status is up to date
16. Gross pay is the total pay for an employee without any deduction
17. Net pay is the pay of an employee after tax deduction
18. Employee salary is an estimate considering the maximum number of hours they can work
19. Discount is applied on adoption fee if the amount is greater than 750 by 5% and when 1000 by 10%
20. Surrender reason cannot exceed 20 characters
21. Date of birth of an animal cannot be greater than date of acquisition

Screenshots of PL/SQL code

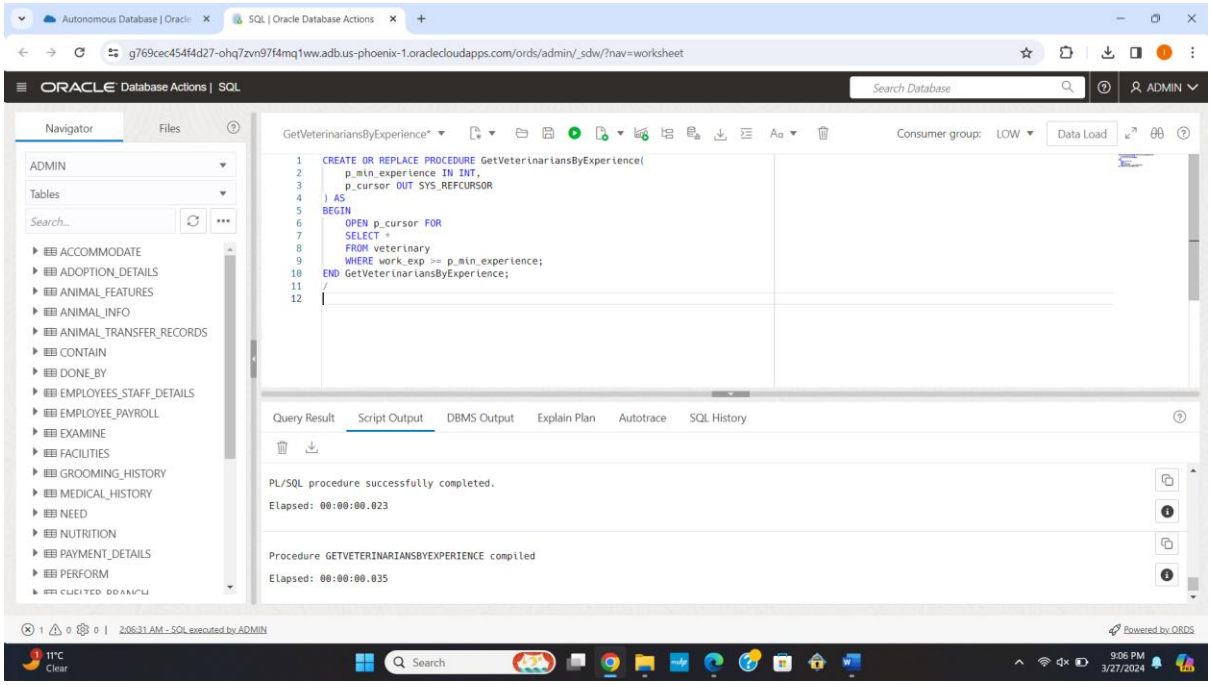
Procedures

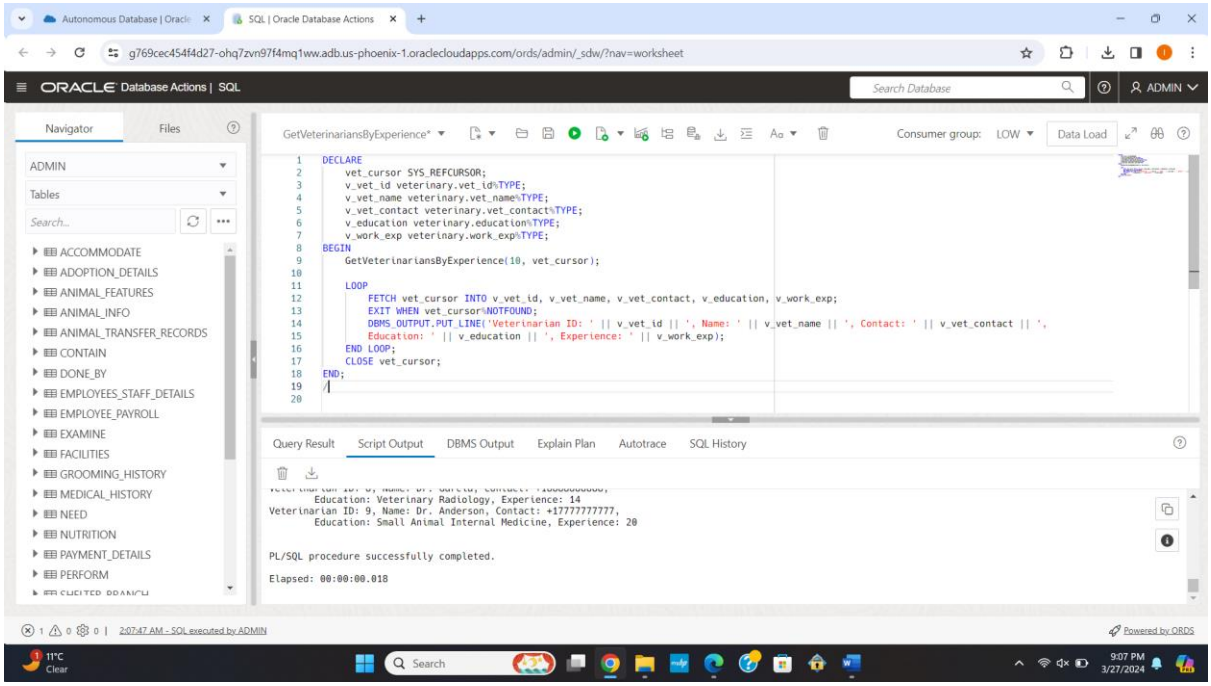
1] Creating a procedure to update the food quantity for animals with up to date vaccination status

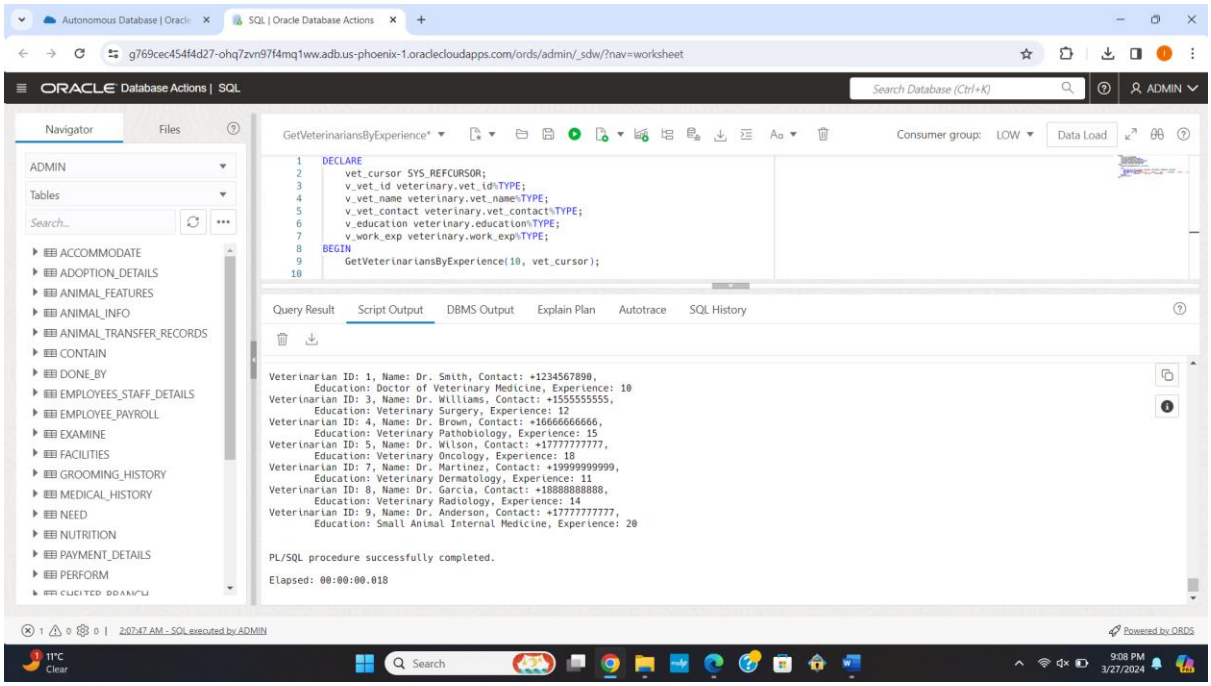




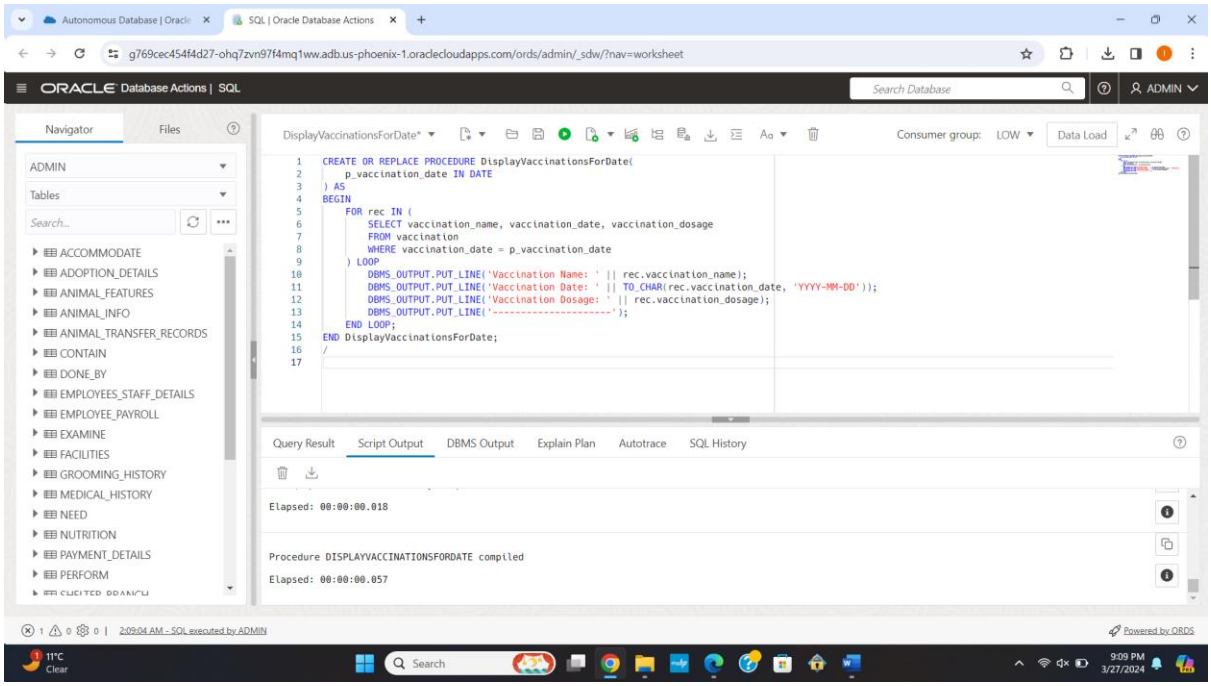
2] Creating a procedure to get the details of Veterinarians with more than x years of work experience

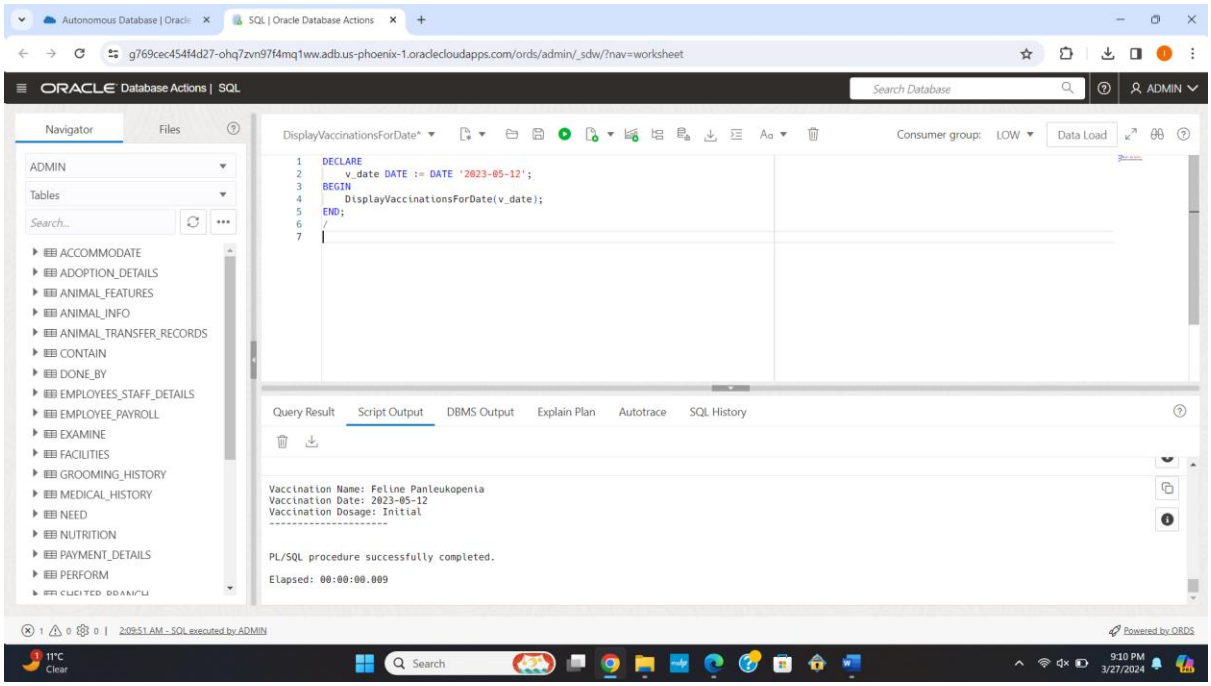




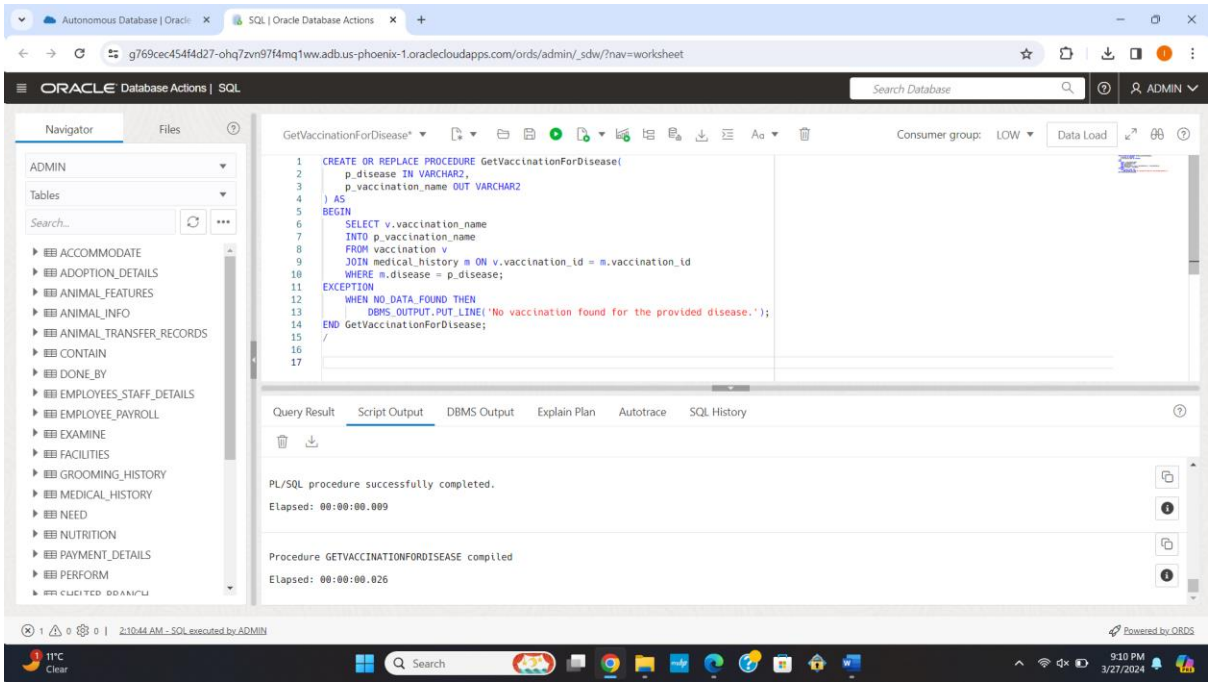


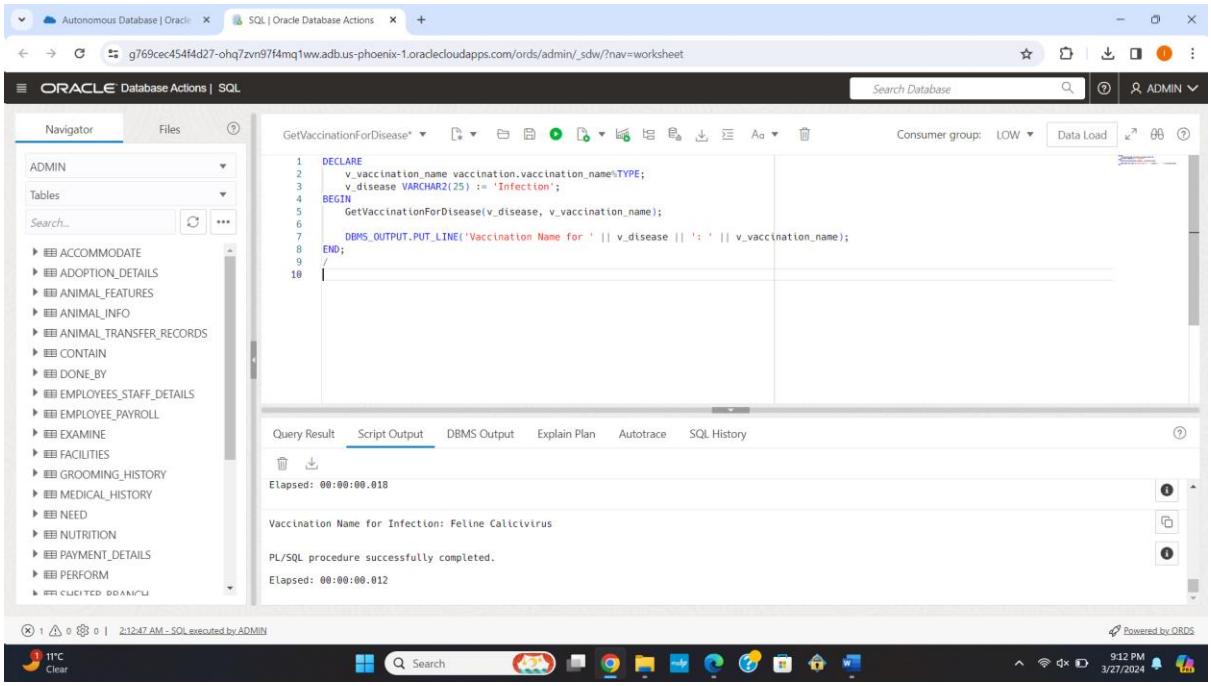
3] Creating a procedure to display the vaccinations given on a particular date



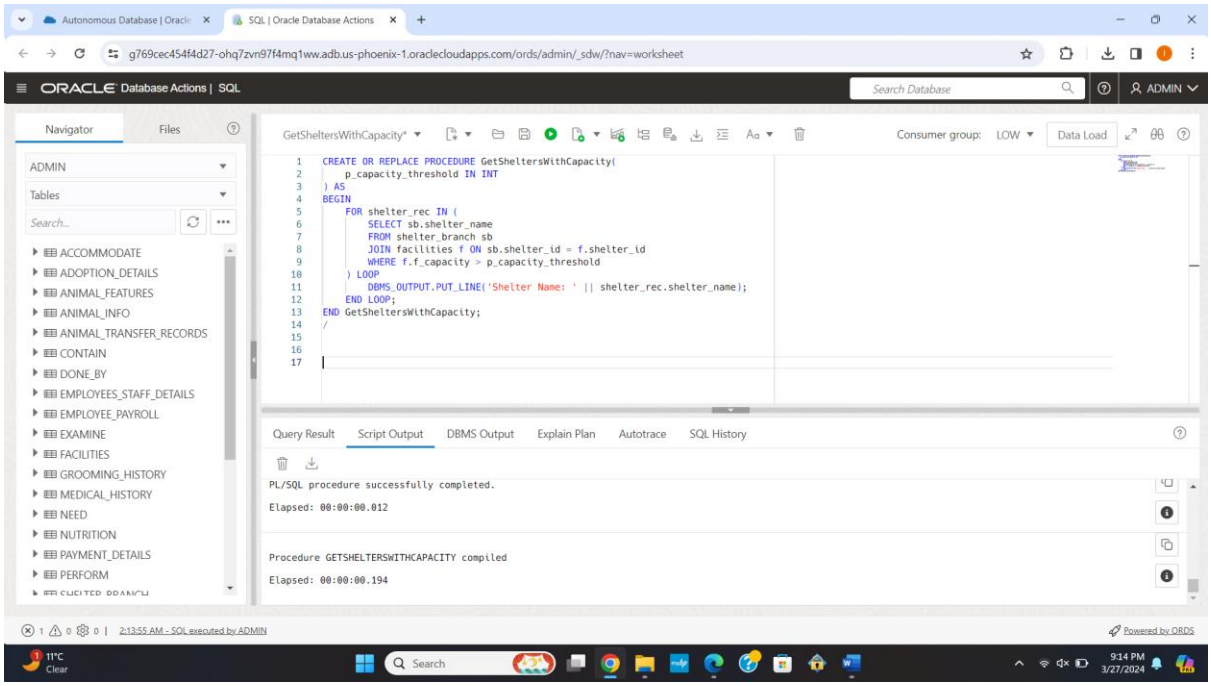


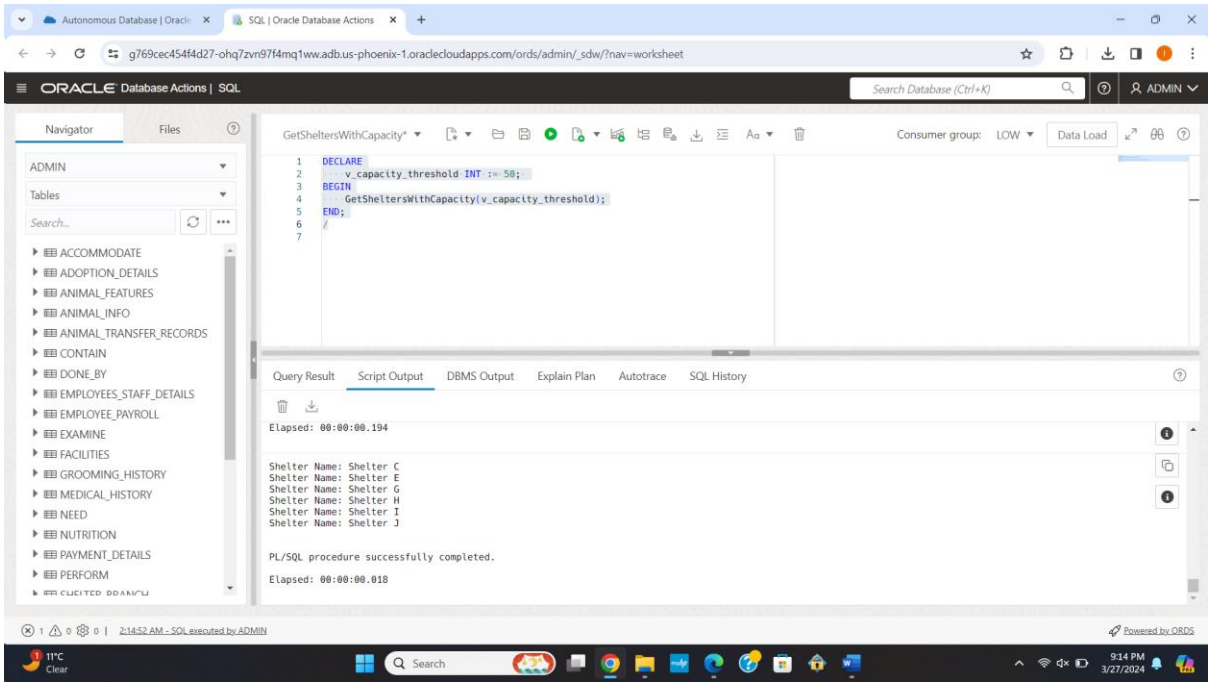
4] Creating a procedure to get the vaccination name for a particular disease as per historical records.





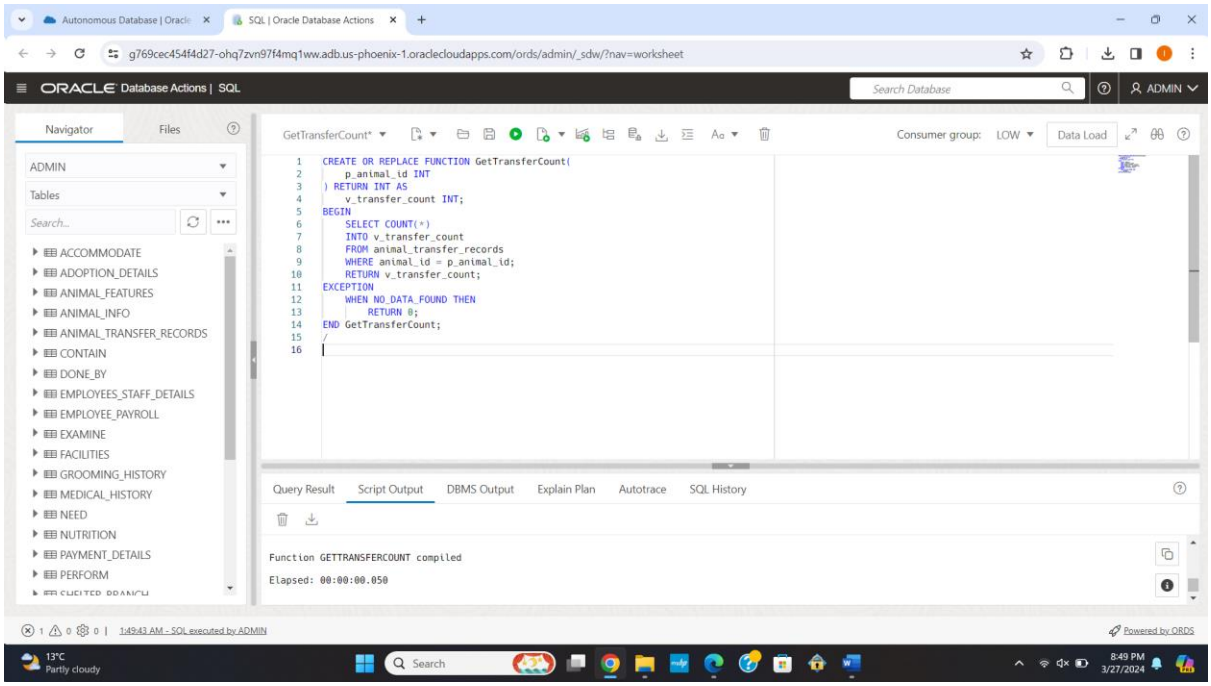
5] Creating a procedure to get the list of all the shelters with capacity more than the given number

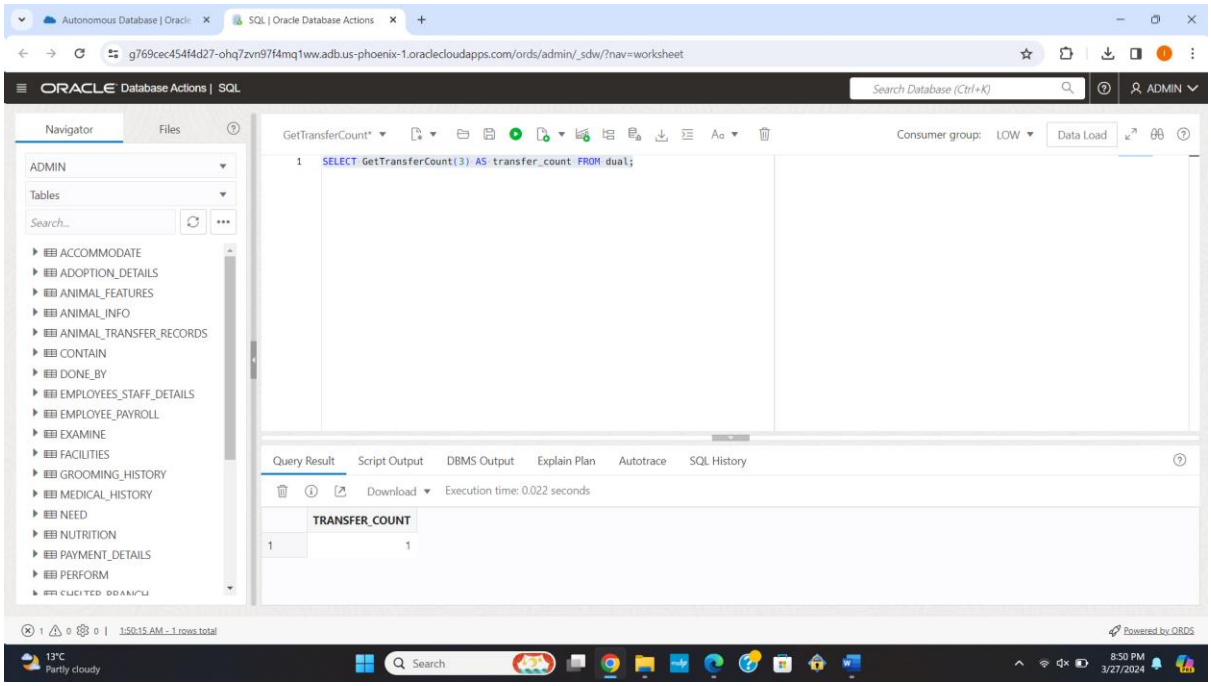




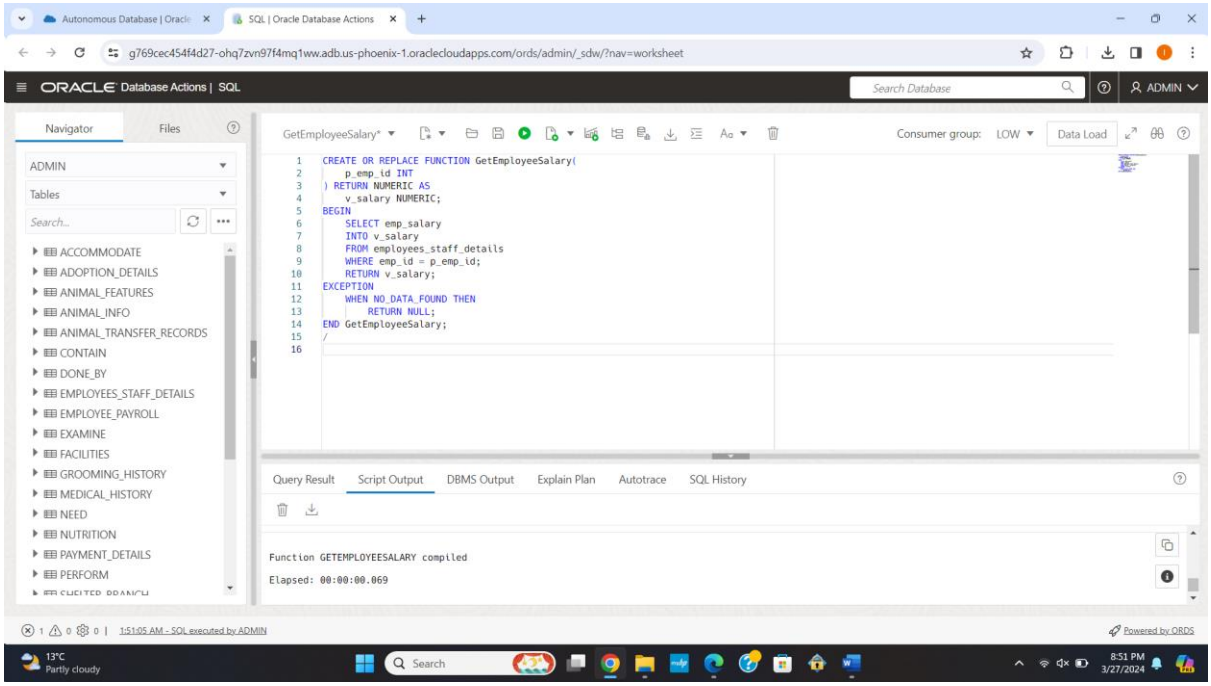
Functions:

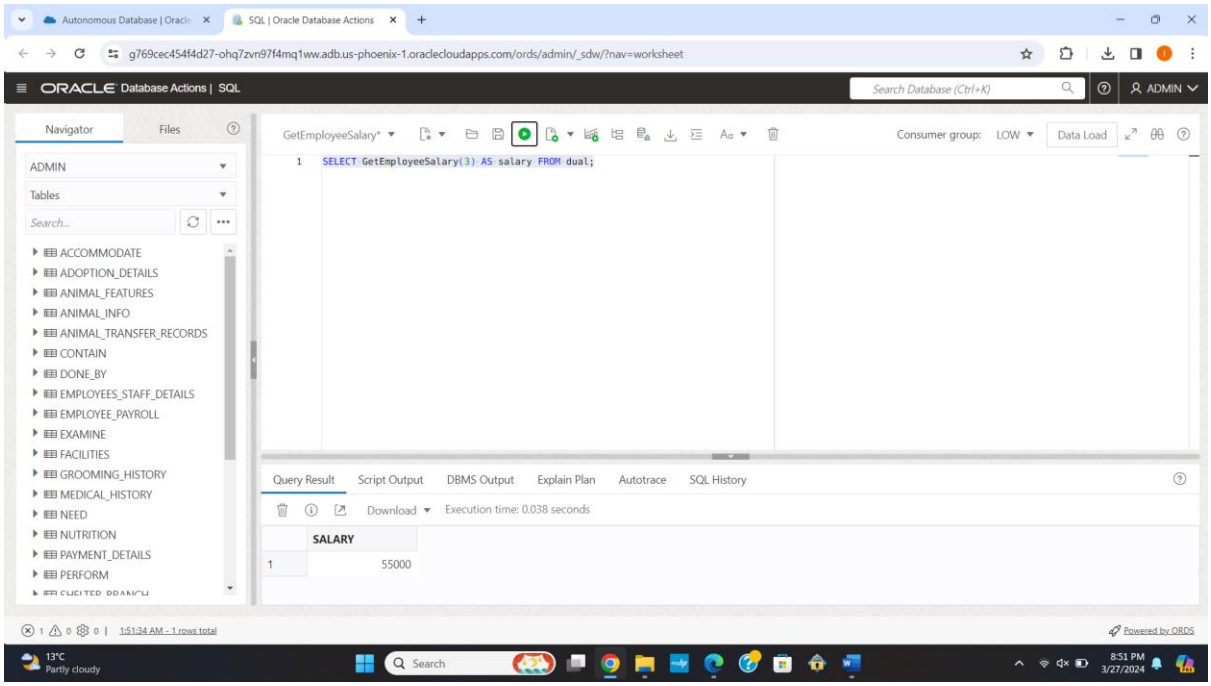
1] Creating a function to get the count of the number of times an animal is transferred from one shelter to another.



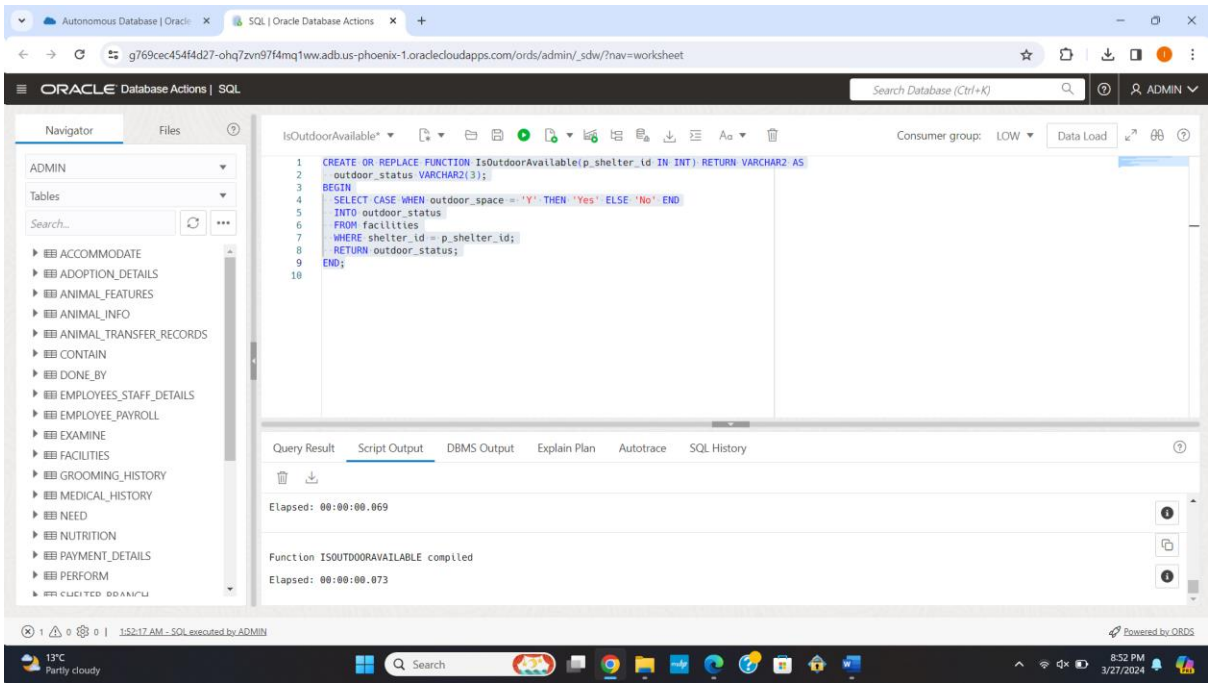


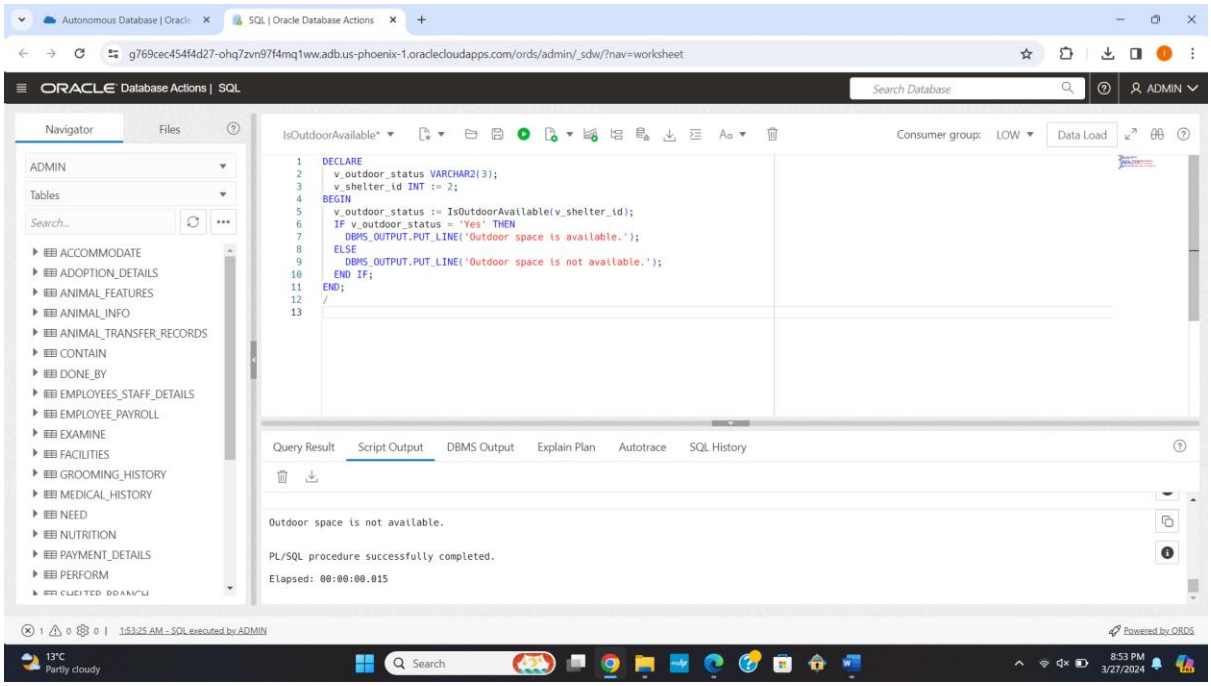
2] Creating a function to get the salary of a given employee



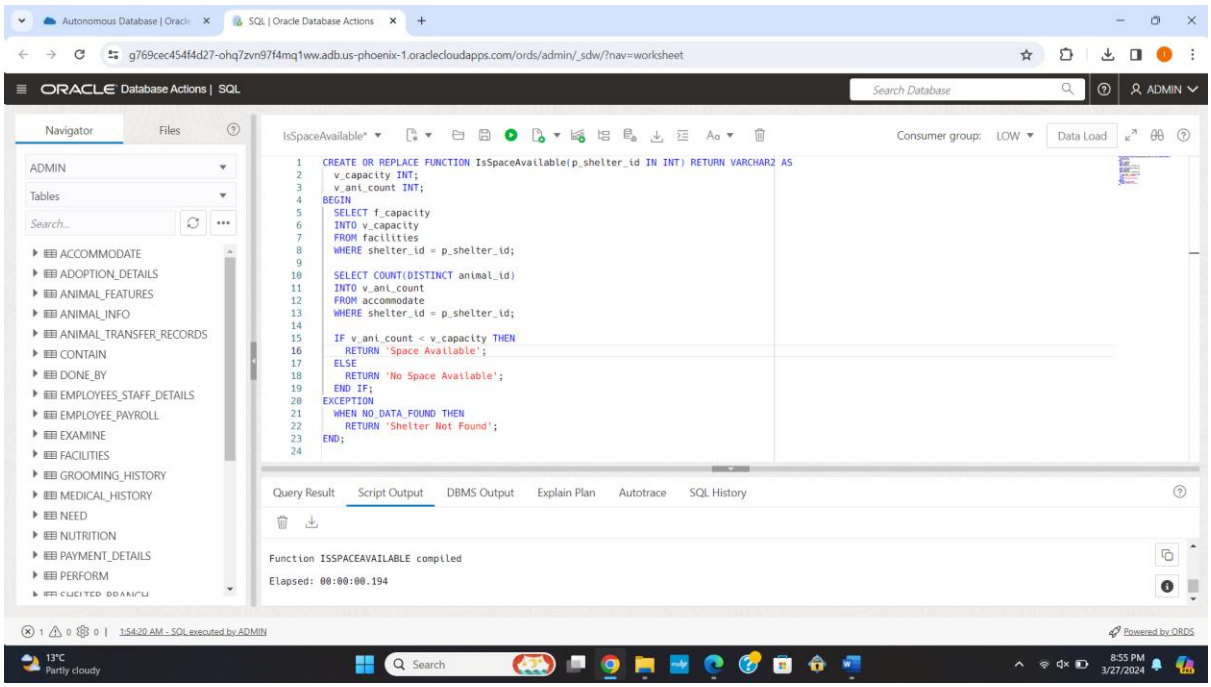


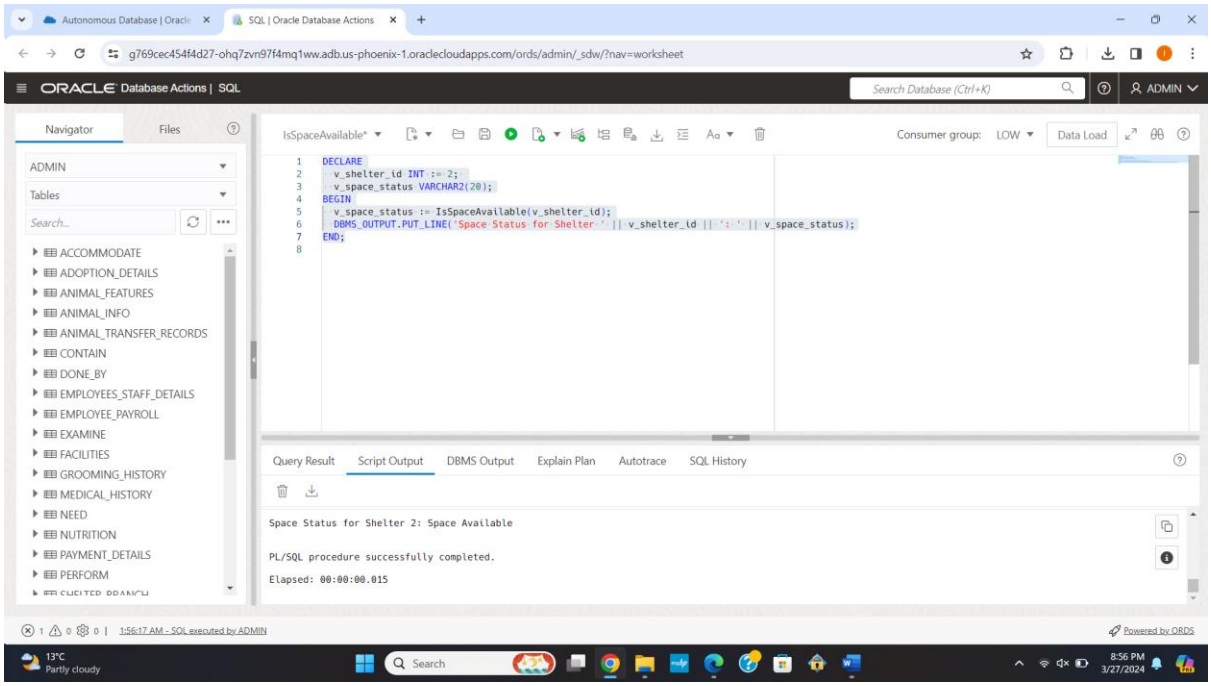
3] Creating a function to check where outdoor space is available for a given shelter ID



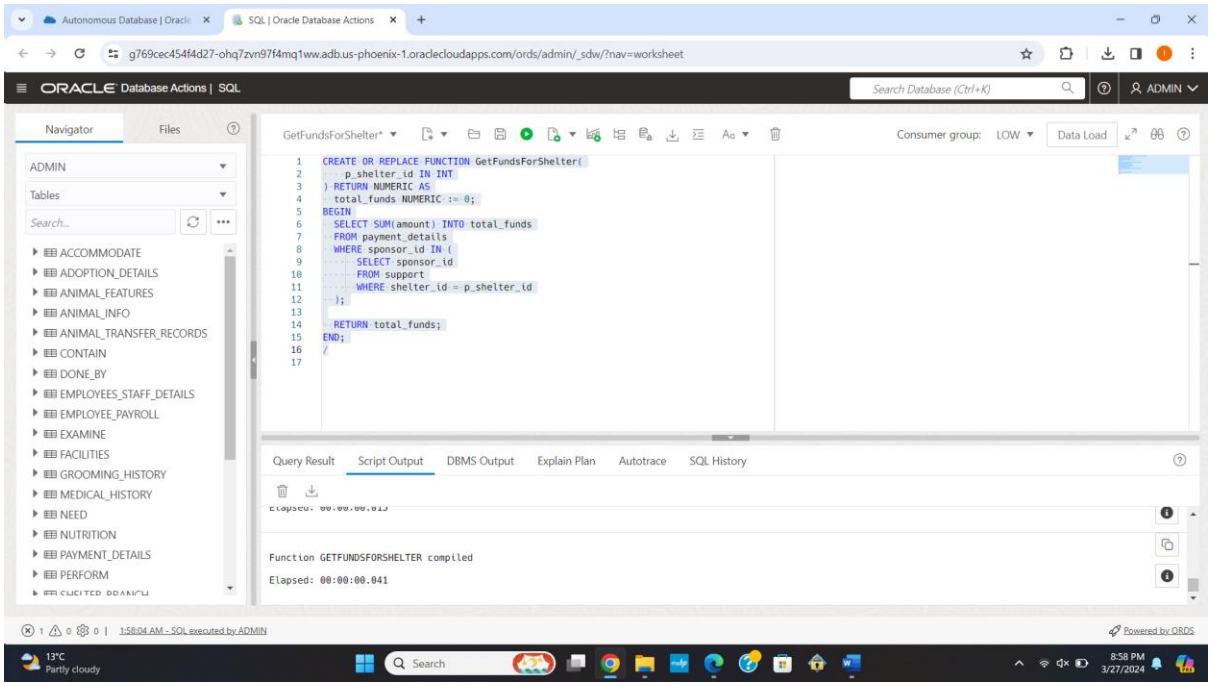


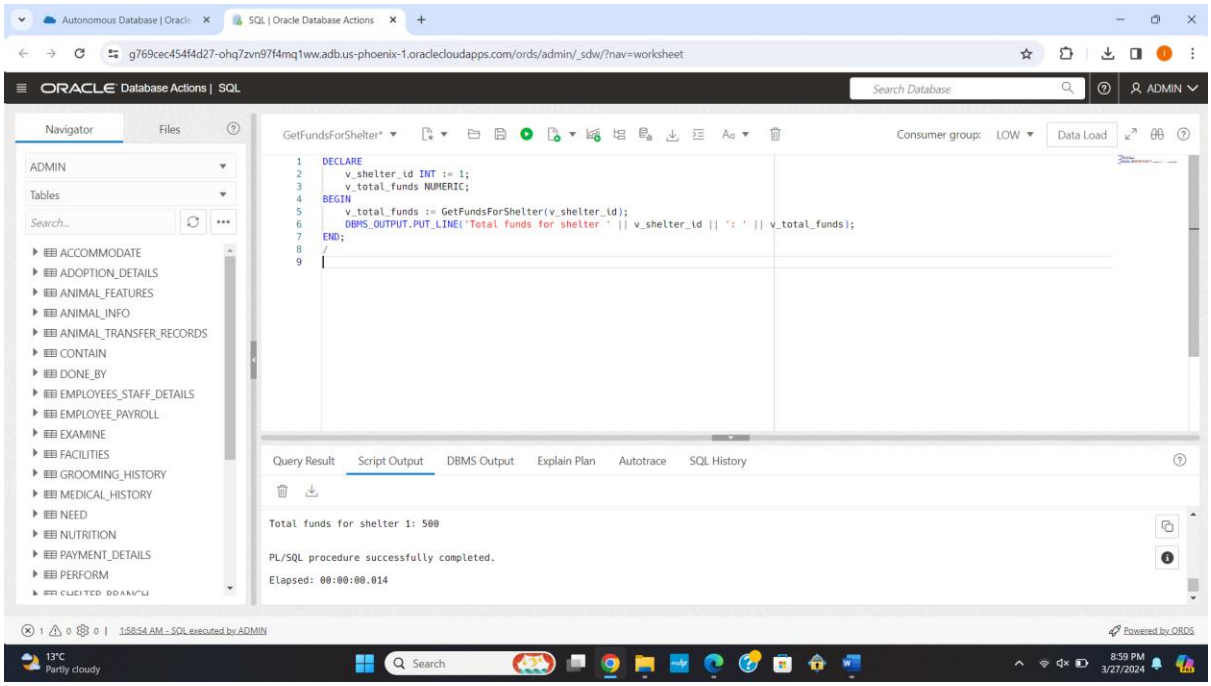
4] Creating a function to check whether a new animal can be inserted into a shelter considering its capacity according to the facilities.





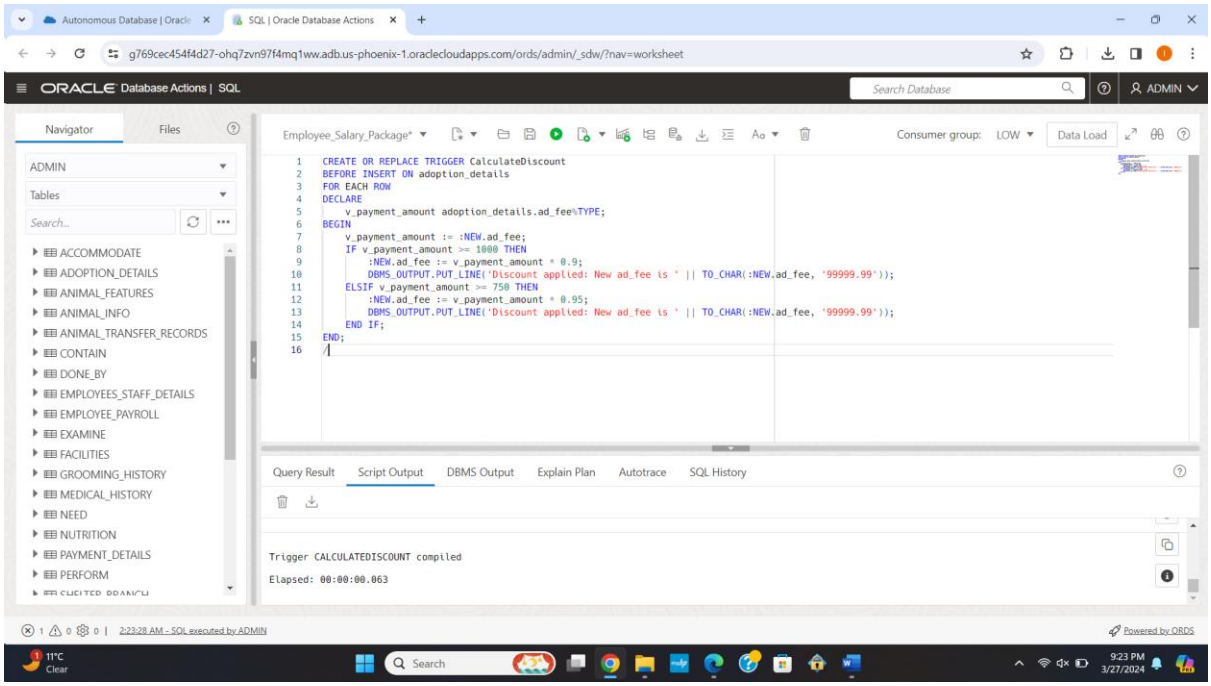
5] Creating a function to get the total funds from the sponsors for a particular shelter ID.

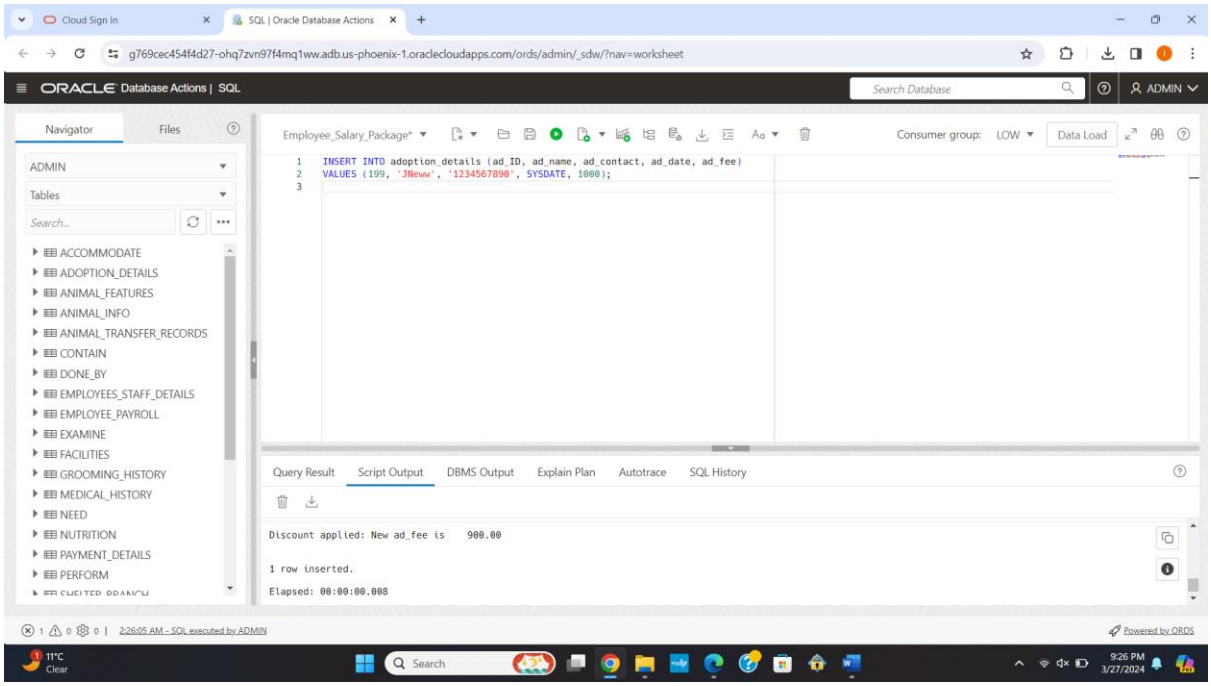




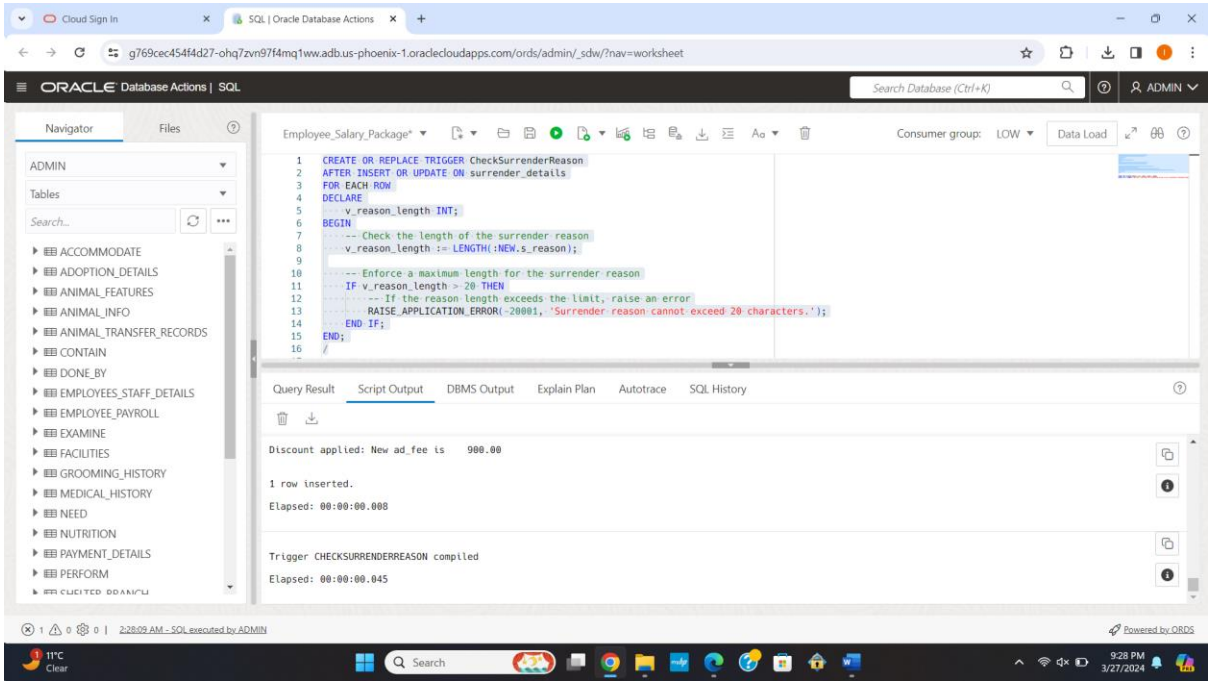
Triggers

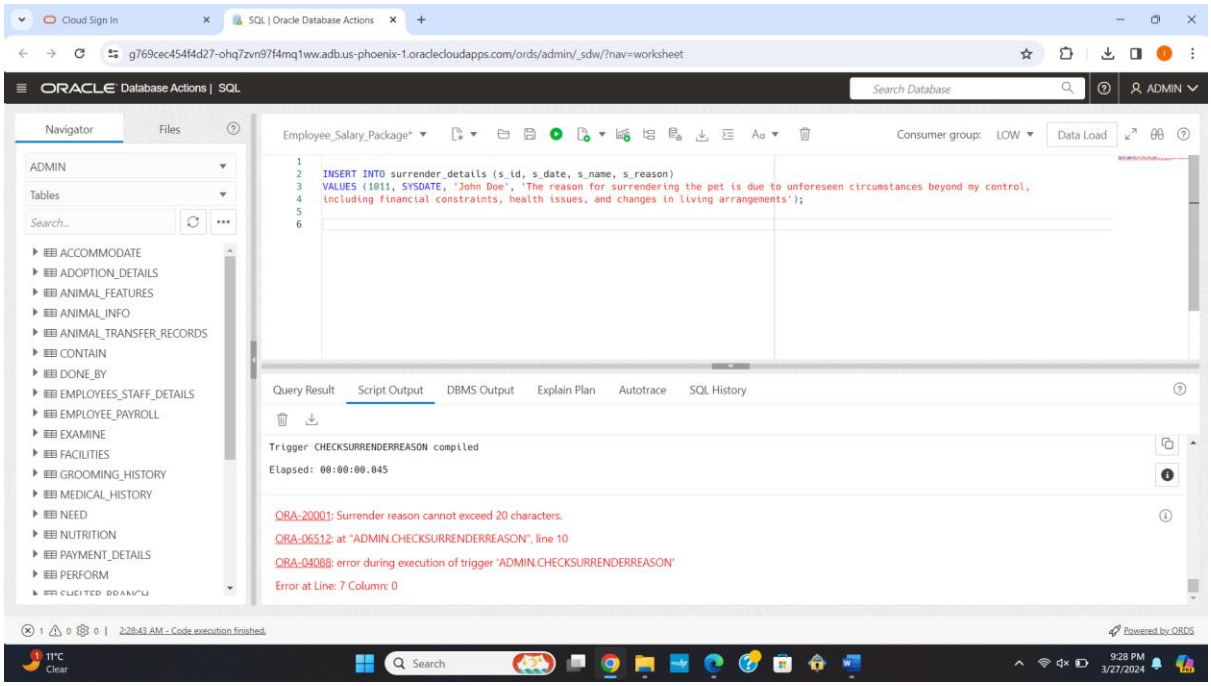
1] Creating a trigger to calculate the adoption fee post discount (10% if 1000 or above, 5% if 750 or above) and update the value in table accordingly.



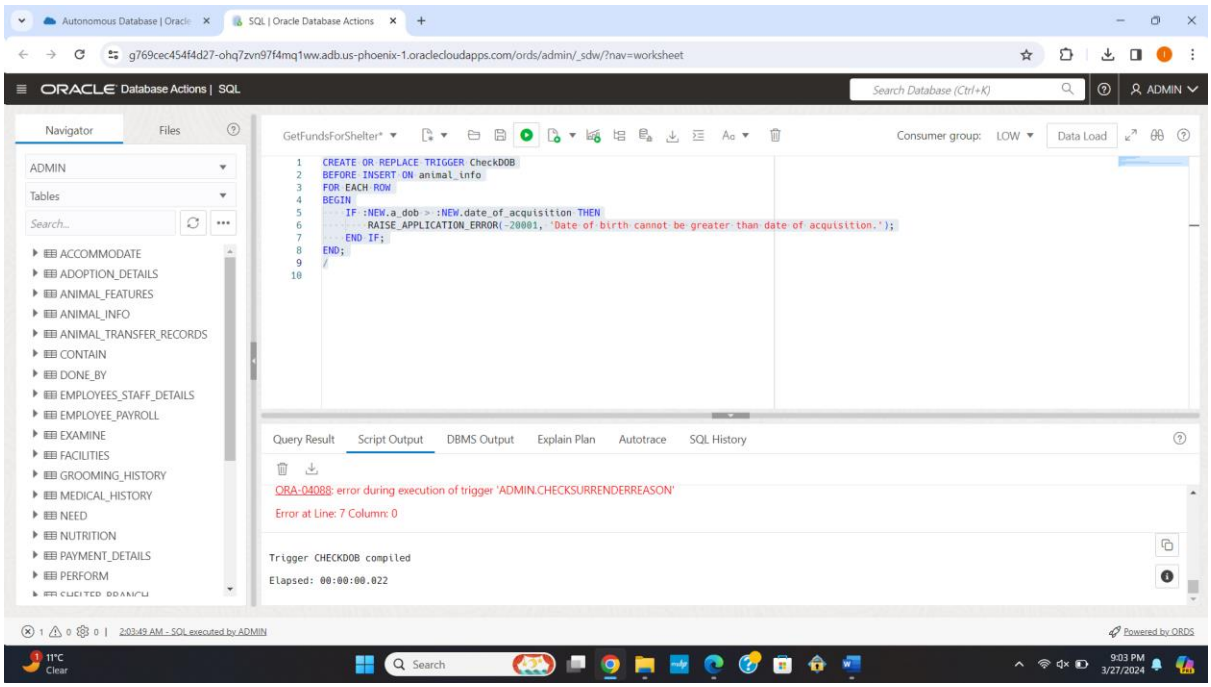


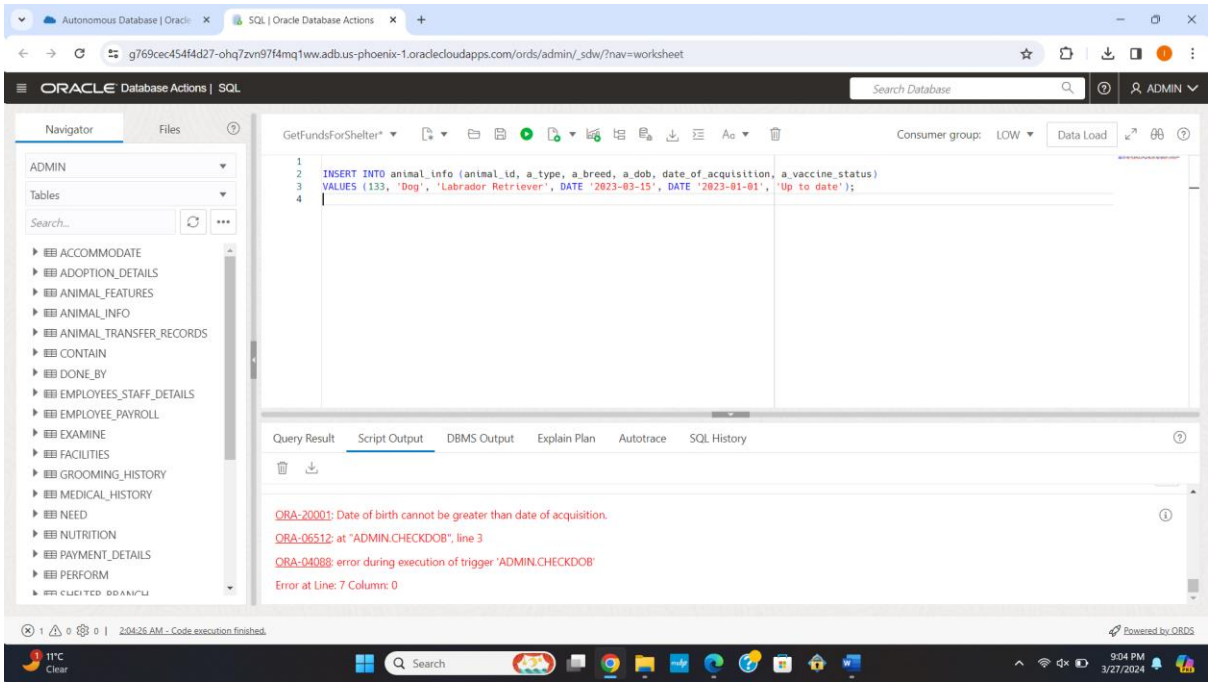
2] Creating a trigger to ensure that the reason for surrendering an animal does not exceed more than 20 characters.





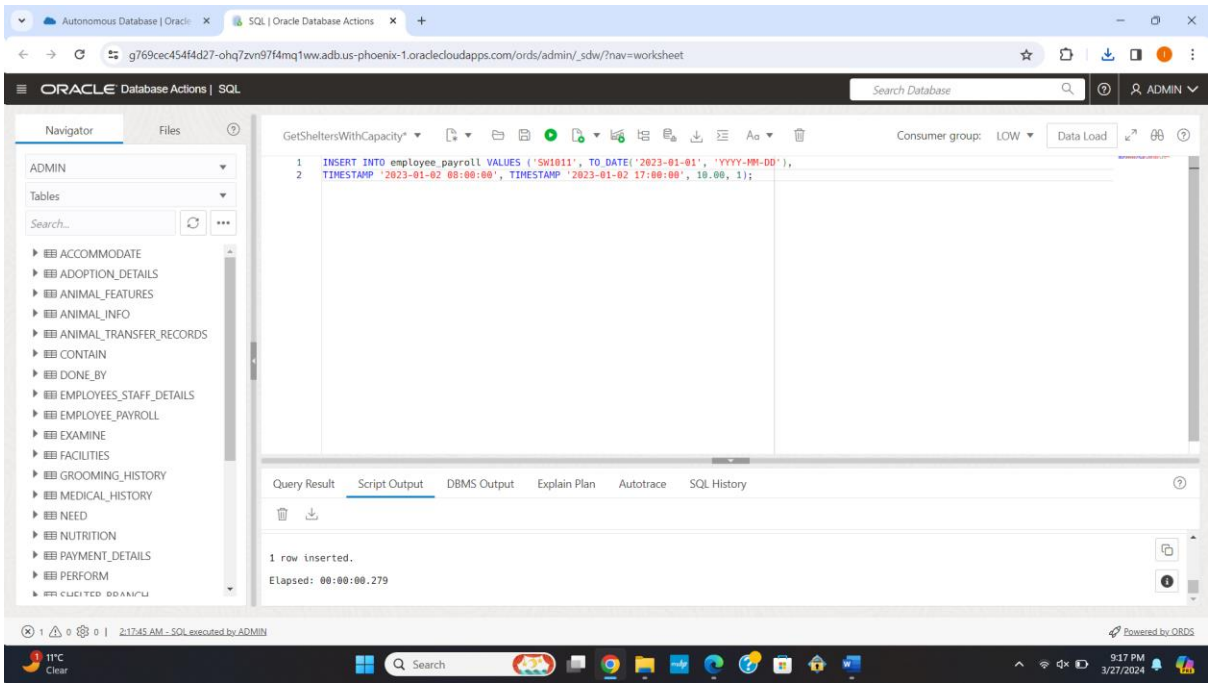
3] Creating a trigger to ensure that the date of acquisition of an animal is always after the date of birth of the animal.

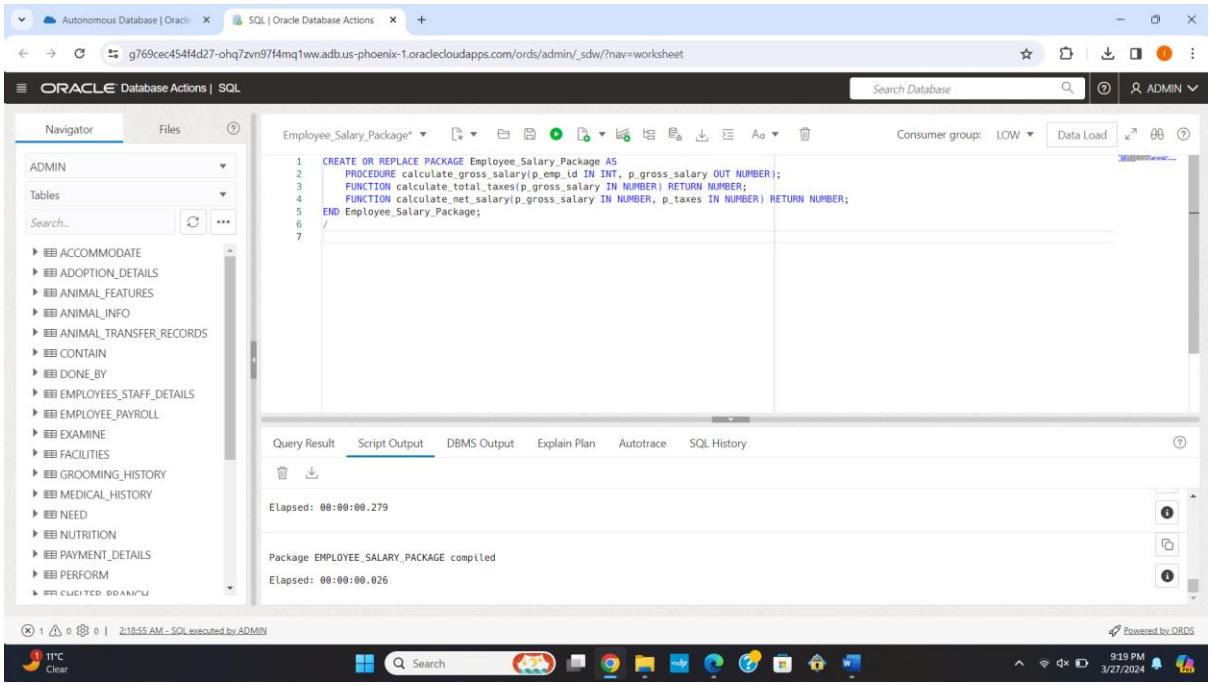


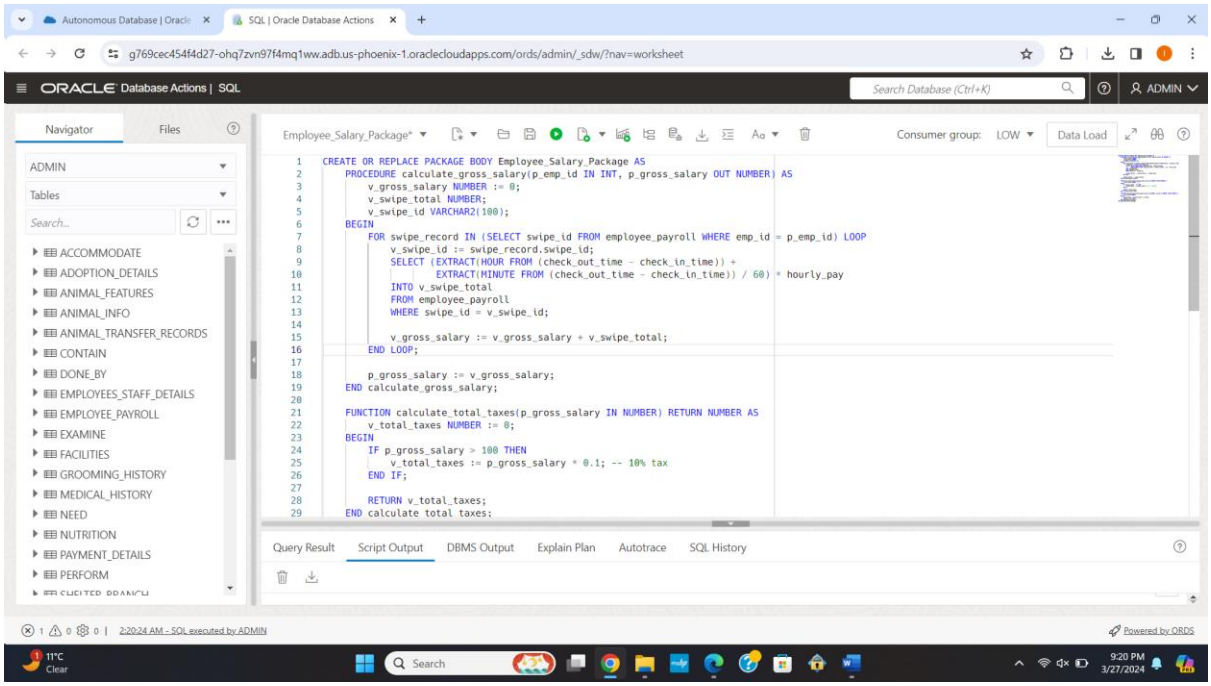


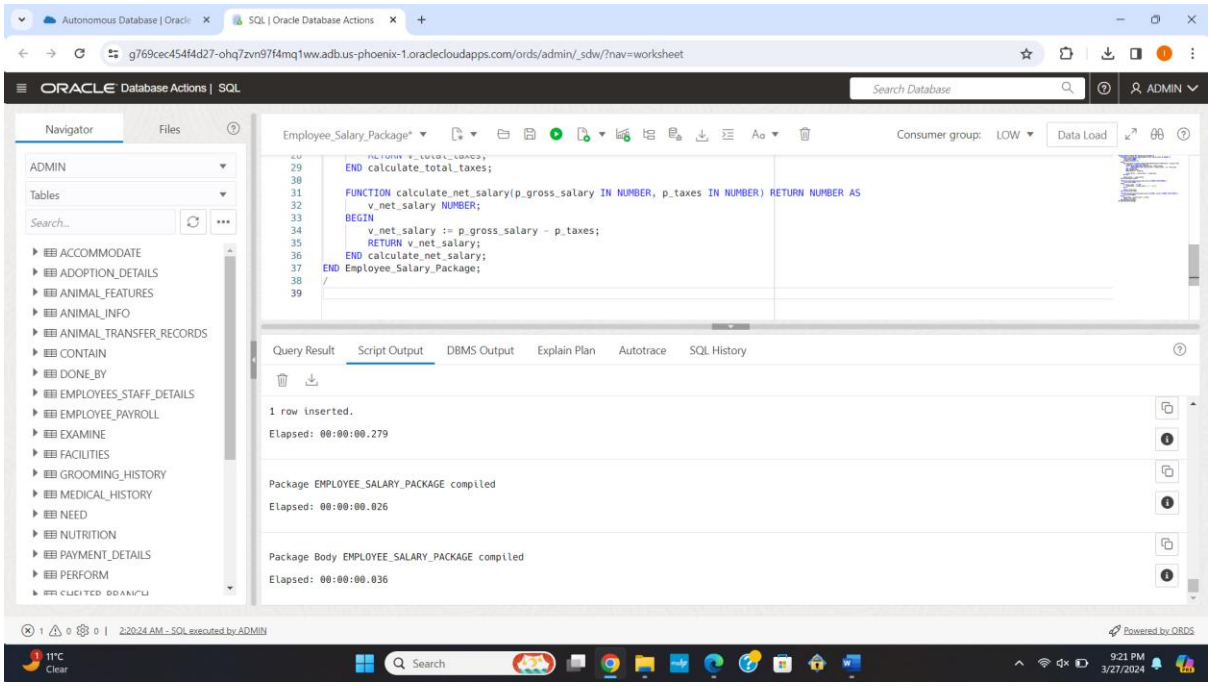
Package

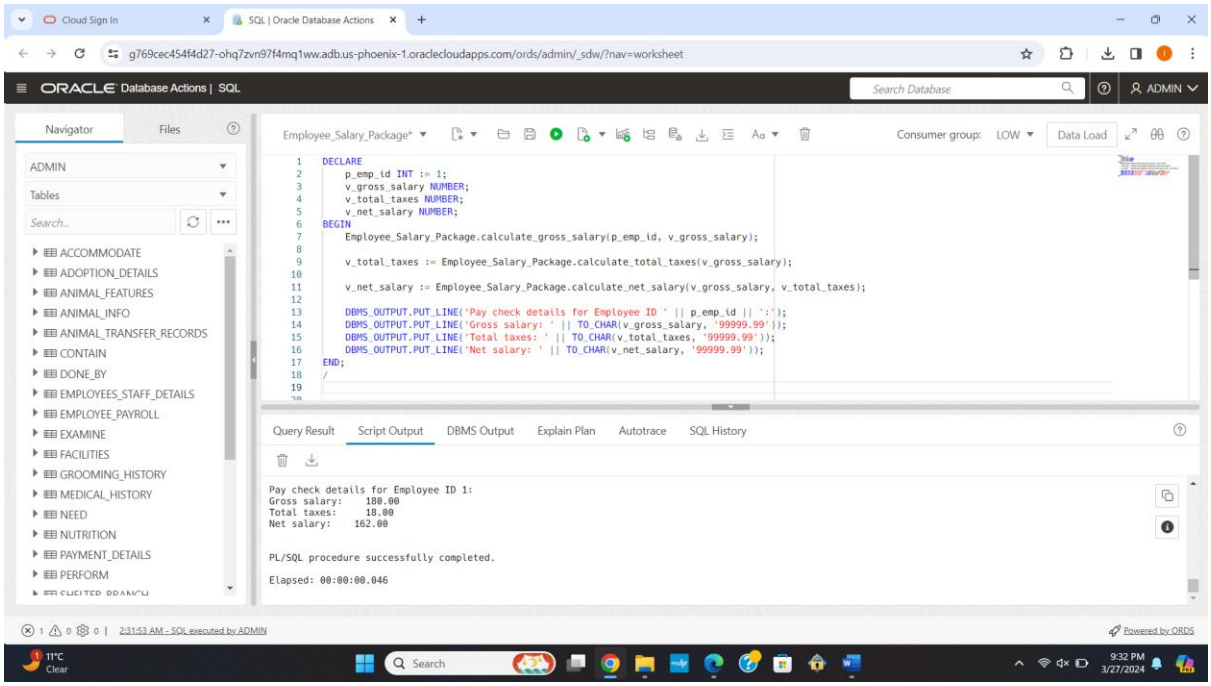
1] Creating a package to calculate the total pay and taxes considering the number of hours worked.




Autonomous Database | Oracle
SQL | Oracle Database Actions
g769cec454f4d27-ohq7zvn97f4mq1ww.adb.us-phoenix-1.oraclecloudapps.com/ords/admin/_sdw/?nav=worksheet
ORACLE Database Actions | SQL
Search Database
ADMIN
Navigator
Files
ADMIN
Tables
Search...
ACCOMMODATE
ADOPTION_DETAILS
ANIMAL_FEATURES
ANIMAL_INFO
ANIMAL_TRANSFER_RECORDS
CONTAIN
DONE_BY
EMPLOYEES_STAFF_DETAILS
EMPLOYEE_PAYROLL
EXAMINE
FACILITIES
GROOMING_HISTORY
MEDICAL_HISTORY
NEED
NUTRITION
PAYMENT_DETAILS
PERFORM
Employee_Salary_Package*
Consumer group: LOW
Data Load
1 CREATE OR REPLACE PACKAGE Employee_Salary_Package AS
2 PROCEDURE calculate_gross_salary(p_emp_id IN INT, p_gross_salary OUT NUMBER);
3 FUNCTION calculate_total_taxes(p_gross_salary IN NUMBER) RETURN NUMBER;
4 FUNCTION calculate_net_salary(p_gross_salary IN NUMBER, p_taxes IN NUMBER) RETURN NUMBER;
5 END Employee_Salary_Package;
6 /
7
Query Result
Script Output
DBMS Output
Explain Plan
Autotrace
SQL History
Elapsed: 00:00:00.279
Package EMPLOYEE_SALARY_PACKAGE compiled
Elapsed: 00:00:00.026
2:18:55 AM - SQL executed by ADMIN
Powered by ORDS
11°C
Clear
Search
9:19 PM
3/27/2024


Autonomous Database | Oracle
SQL | Oracle Database Actions
g769cec454f4d27-ohq7zvn97f4mq1ww.adb.us-phoenix-1.oraclecloudapps.com/ords/admin/_sdw/?nav=worksheet
ORACLE Database Actions | SQL
Search Database (Ctrl+K)
ADMIN
Navigator
Files
ADMIN
Tables
Search...
ACCOMMODATE
ADOPTION_DETAILS
ANIMAL_FEATURES
ANIMAL_INFO
ANIMAL_TRANSFER_RECORDS
CONTAIN
DONE_BY
EMPLOYEES_STAFF_DETAILS
EMPLOYEE_PAYROLL
EXAMINE
FACILITIES
GROOMING_HISTORY
MEDICAL_HISTORY
NEED
NUTRITION
PAYMENT_DETAILS
PERFORM
Employee_Salary_Package*
Consumer group: LOW
Data Load
1 CREATE OR REPLACE PACKAGE BODY Employee_Salary_Package AS
2 PROCEDURE calculate_gross_salary(p_emp_id IN INT, p_gross_salary OUT NUMBER) AS
3 v_gross_salary NUMBER := 0;
4 v_swipe_total NUMBER;
5 v_swipe_id VARCHAR2(100);
6 BEGIN
7 FOR swipe_record IN (SELECT swipe_id FROM employee_payroll WHERE emp_id = p_emp_id) LOOP
8 v_swipe_id := swipe_record.swipe_id;
9 SELECT (EXTRACT(HOUR FROM (check_out_time - check_in_time)) +
10 EXTRACT(MINUTE FROM (check_out_time - check_in_time)) / 60) * hourly_pay
11 INTO v_swipe_total
12 FROM employee_payroll
13 WHERE swipe_id = v_swipe_id;
14
15 v_gross_salary := v_gross_salary + v_swipe_total;
16 END LOOP;
17
18 p_gross_salary := v_gross_salary;
19 END calculate_gross_salary;
20
21 FUNCTION calculate_total_taxes(p_gross_salary IN NUMBER) RETURN NUMBER AS
22 v_total_taxes NUMBER := 0;
23 BEGIN
24 IF p_gross_salary > 100 THEN
25 v_total_taxes := p_gross_salary * 0.1; -- 10% tax
26 END IF;
27
28 RETURN v_total_taxes;
29 END calculate total taxes;
Query Result
Script Output
DBMS Output
Explain Plan
Autotrace
SQL History
2:20:24 AM - SQL executed by ADMIN
Powered by ORDS
11°C
Clear
Search
9:20 PM
3/27/2024


Autonomous Database | Oracle
SQL | Oracle Database Actions
g769cec454f4d27-ohq7zvn97f4mq1ww.adb.us-phoenix-1.oraclecloudapps.com/ords/admin/_sdw/?nav=worksheet
ORACLE Database Actions | SQL
Search Database
ADMIN
Navigator
Files
ADMIN
Tables
Search...
ACCOMMODATE
ADOPTION_DETAILS
ANIMAL_FEATURES
ANIMAL_INFO
ANIMAL_TRANSFER_RECORDS
CONTAIN
DONE_BY
EMPLOYEES_STAFF_DETAILS
EMPLOYEE_PAYROLL
EXAMINE
FACILITIES
GROOMING_HISTORY
MEDICAL_HISTORY
NEED
NUTRITION
PAYMENT_DETAILS
PERFORM
Employee_Salary_Package*
Consumer group: LOW
Data Load
END calculate_total_taxes;
FUNCTION calculate_net_salary(p_gross_salary IN NUMBER, p_taxes IN NUMBER) RETURN NUMBER AS
v_net_salary NUMBER;
BEGIN
v_net_salary := p_gross_salary - p_taxes;
RETURN v_net_salary;
END calculate_net_salary;
END Employee_Salary_Package;
/
Query Result
Script Output
DBMS Output
Explain Plan
Autotrace
SQL History
1 row inserted.
Elapsed: 00:00:00.279
Package EMPLOYEE_SALARY_PACKAGE compiled
Elapsed: 00:00:00.026
Package Body EMPLOYEE_SALARY_PACKAGE compiled
Elapsed: 00:00:00.036
2:20:24 AM - SQL executed by ADMIN
Powered by ORDS
11°C
Clear
Search
9:21 PM
3/27/2024


Cloud Sign In
SQL | Oracle Database Actions
g769cec454f4d27-ohq7zvn97f4mq1ww.adb.us-phoenix-1.oraclecloudapps.com/ords/admin/_sdw/?nav=worksheet
ORACLE Database Actions | SQL
Search Database
ADMIN
Navigator
Files
ADMIN
Tables
Search...
ACCOMMODATE
ADOPTION_DETAILS
ANIMAL_FEATURES
ANIMAL_INFO
ANIMAL_TRANSFER_RECORDS
CONTAIN
DONE_BY
EMPLOYEES_STAFF_DETAILS
EMPLOYEE_PAYROLL
EXAMINE
FACILITIES
GROOMING_HISTORY
MEDICAL_HISTORY
NEED
NUTRITION
PAYMENT_DETAILS
PERFORM
Employee_Salary_Package*
Consumer group: LOW
Data Load
DECLARE
p_emp_id INT := 1;
v_gross_salary NUMBER;
v_total_taxes NUMBER;
v_net_salary NUMBER;
BEGIN
Employee_Salary_Package.calculate_gross_salary(p_emp_id, v_gross_salary);
v_total_taxes := Employee_Salary_Package.calculate_total_taxes(v_gross_salary);
v_net_salary := Employee_Salary_Package.calculate_net_salary(v_gross_salary, v_total_taxes);
DBMS_OUTPUT.PUT_LINE('Pay check details for Employee ID ' || p_emp_id || ':');
DBMS_OUTPUT.PUT_LINE('Gross salary: ' || TO_CHAR(v_gross_salary, '99999.99'));
DBMS_OUTPUT.PUT_LINE('Total taxes: ' || TO_CHAR(v_total_taxes, '99999.99'));
DBMS_OUTPUT.PUT_LINE('Net salary: ' || TO_CHAR(v_net_salary, '99999.99'));
END;
/
Query Result
Script Output
DBMS Output
Explain Plan
Autotrace
SQL History
Pay check details for Employee ID 1:
Gross salary: 180.00
Total taxes: 18.00
Net salary: 162.00
PL/SQL procedure successfully completed.
Elapsed: 00:00:00.046
2:31:53 AM - SQL executed by ADMIN
Powered by ORDS
11°C
Clear
Search
9:32 PM
3/27/2024

Individual Contribution:

The intention of this part was to develop an electronic environment that would help the shelter very purpose to save the animals and the personnel looking after them, rather than developing just a database.

This is the part where the ‘GetFundsForShelter’ function has been created. In line with being an important part of making sure financial accountability and transparency are adhered to, I have designed it in a manner where it can perfectly keep track and systematically record all amounts received for each shelter under its sponsors. This was a way to ensure that each donation went straight into being able to better the life of the animals rather than a way of fundraising. Accordingly, the function was written correctly, tested rigorously, and used by others, including this, to become a reliable source in maintaining and helping to create a supportive and trusting environment of personnel and contributors of the shelter.

I developed the 'GetSheltersWithCapacity' Procedure focusing on the operating features of the shelters. To be precise, it aims at enabling a simplified yet effective way of the improved resources' allocation, operational planning, and identified shelters that over a certain limit barrier therein. It is my real hope that this information will be able to create a true bridge between data administration and practical parts of shelter life, so that every shelter will have a possibility of working within its means while still maintaining the highest standards of animal care.

The creation of the ‘CheckDOB’ trigger was perhaps the most meaningful example of my commitment to the part. Realization made on the above shortcoming instigated the creation of this trigger to ensure the integrity of the animals' records, especially the date of acquisition being logically sequenced after the date of birth, given the basic significance of correct record-keeping in animal welfare. This feature of the database system does show a deep respect for the travels and pasts of these animals to ensure their care began on a foundation of truth and trust. It confirmed for me that I can use technology to improve and safeguard the lives of those under our care. Each trigger, procedure, and function developed was goal-centric and commitment had "commitment" as a word for seeing measurable improvement in the shelter community.