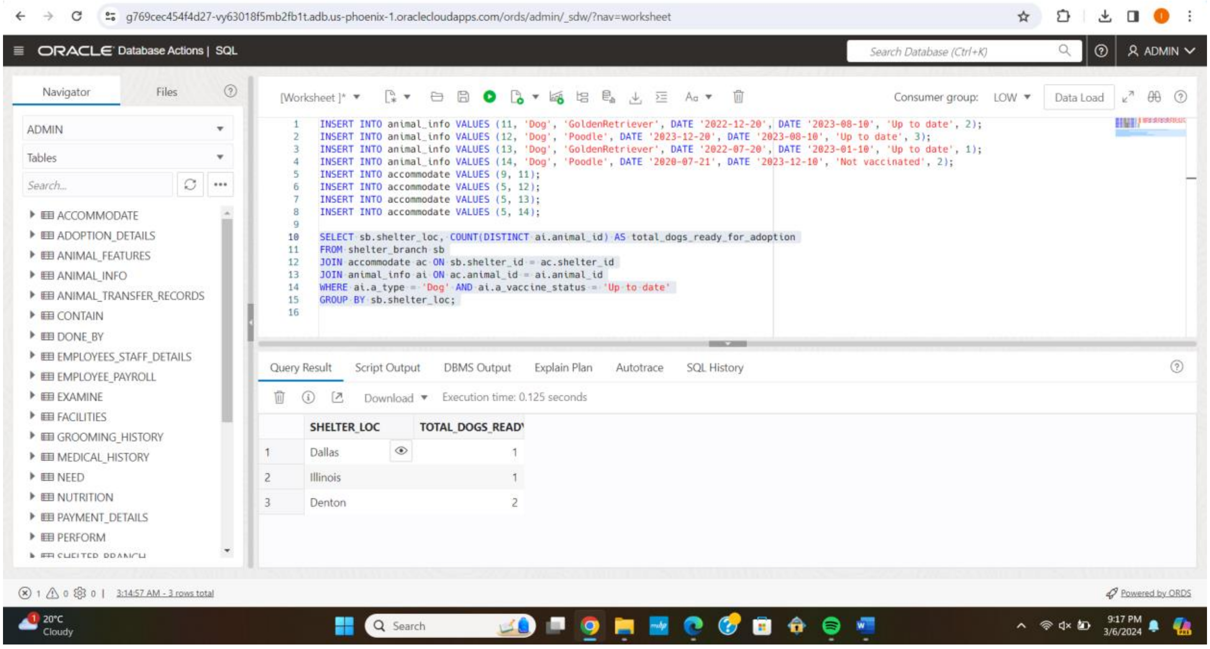
PROJECT PART 4

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

Given Queries:

1. List the total number of dogs ready for adoption by location.



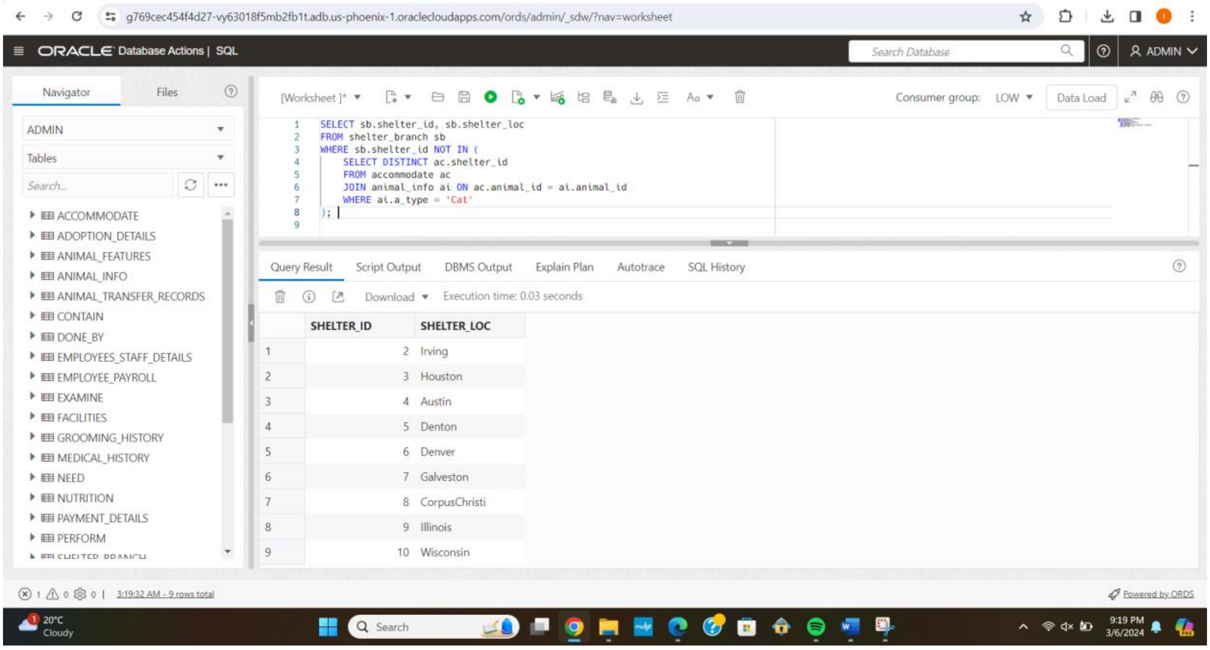
```sql
INSERT INTO animal_info VALUES (11, 'Dog', 'GoldenRetriever', DATE '2022-12-20', DATE '2023-08-10', 'Up to date', 2);
INSERT INTO animal_info VALUES (12, 'Dog', 'Poodle', DATE '2023-12-20', DATE '2023-08-10', 'Up to date', 3);
INSERT INTO animal_info VALUES (13, 'Dog', 'GoldenRetriever', DATE '2022-07-20', DATE '2023-01-10', 'Up to date', 1);
INSERT INTO animal_info VALUES (14, 'Dog', 'Poodle', DATE '2020-07-21', DATE '2023-12-10', 'Not vaccinated', 2);
INSERT INTO accommodate VALUES (9, 11);
INSERT INTO accommodate VALUES (5, 12);
INSERT INTO accommodate VALUES (5, 13);
INSERT INTO accommodate VALUES (5, 14);

SELECT sb.shelter_loc, COUNT(DISTINCT ai.animal_id) AS total_dogs_ready_for_adoption
FROM shelter_branch sb
JOIN accommodate ac ON sb.shelter_id = ac.shelter_id
JOIN animal_info ai ON ac.animal_id = ai.animal_id
WHERE ai.a_type = 'Dog' AND ai.a_vaccine_status = 'Up to date'
GROUP BY sb.shelter_loc;
```

Execution time: 0.125 seconds

	SHELTER_LOC	TOTAL_DOGS_READY
1	Dallas	1
2	Illinois	1
3	Denton	2

3:14:57 AM - 3 rows total

2. Find animal shelter locations without cats available for adoption:



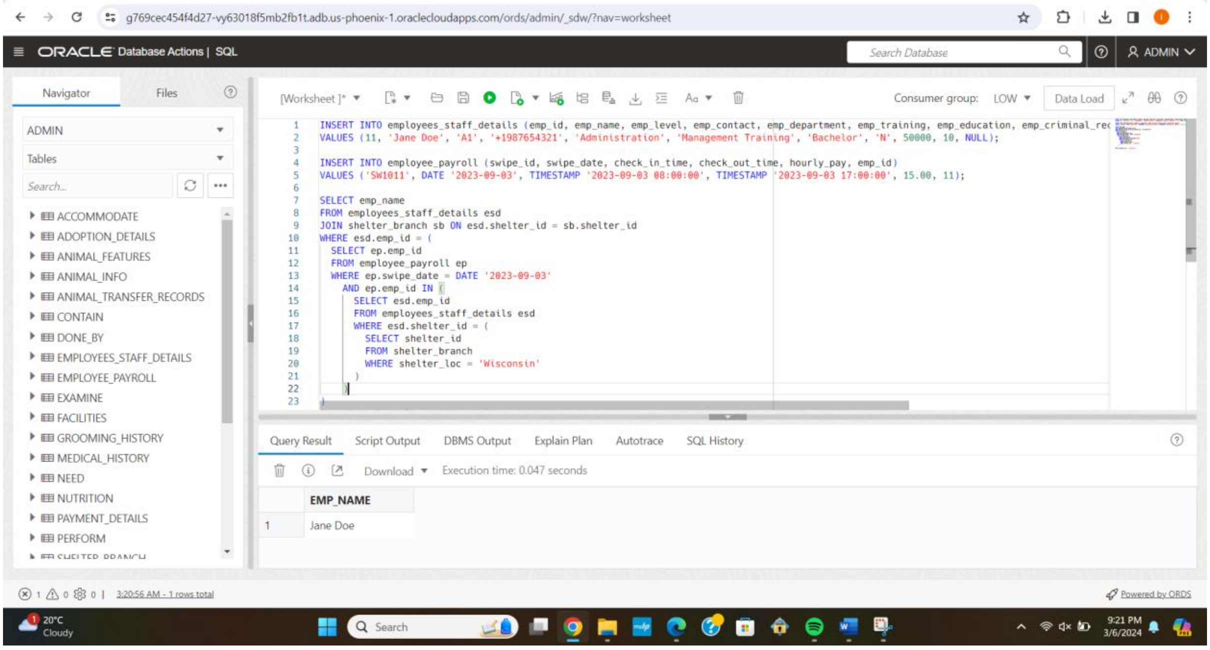
```sql
SELECT sb.shelter_id, sb.shelter_loc
FROM shelter_branch sb
WHERE sb.shelter_id NOT IN (
    SELECT DISTINCT ac.shelter_id
    FROM accommodate ac
    JOIN animal_info ai ON ac.animal_id = ai.animal_id
    WHERE ai.a_type = 'Cat'
);
```

Execution time: 0.03 seconds

	SHELTER_ID	SHELTER_LOC
1	2	Irving
2	3	Houston
3	4	Austin
4	5	Denton
5	6	Denver
6	7	Galveston
7	8	CorpusChristi
8	9	Illinois
9	10	Wisconsin

3:19:32 AM - 9 rows total

3. Find the name of the employee(s) that had worked the most hours on September3, 2023 in a shelter located in Wisconsin:



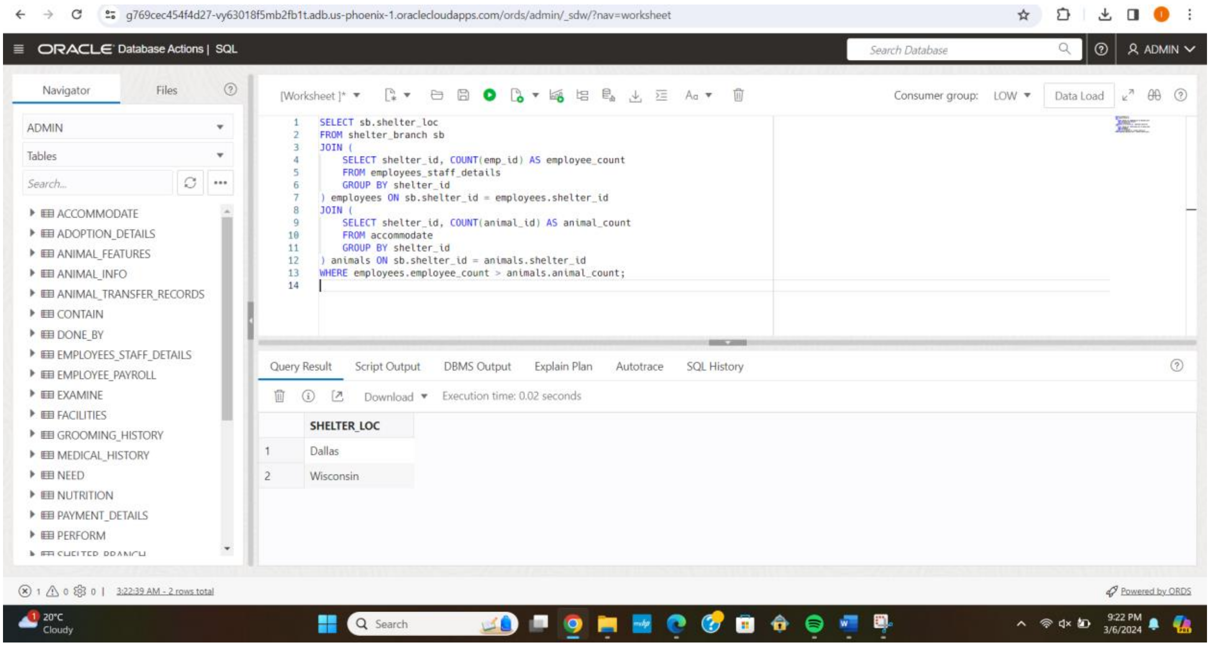
```sql
INSERT INTO employees_staff_details (emp_id, emp_name, emp_level, emp_contact, emp_department, emp_training, emp_education, emp_criminal_rec
VALUES (11, 'Jane Doe', 'A1', '+1987654321', 'Administration', 'Management Training', 'Bachelor', 'N', 50000, 10, NULL);

INSERT INTO employee_payroll (swipe_id, swipe_date, check_in_time, check_out_time, hourly_pay, emp_id)
VALUES ('SW1011', DATE '2023-09-03', TIMESTAMP '2023-09-03 08:00:00', TIMESTAMP '2023-09-03 17:00:00', 15.00, 11);

SELECT emp_name
FROM employees_staff_details esd
JOIN shelter_branch sb ON esd.shelter_id = sb.shelter_id
WHERE esd.emp_id = (
  SELECT ep.emp_id
  FROM employee_payroll ep
  WHERE ep.swipe_date = DATE '2023-09-03'
    AND ep.emp_id IN (
      SELECT esd.emp_id
      FROM employees_staff_details esd
      WHERE esd.shelter_id = (
        SELECT shelter_id
        FROM shelter_branch
        WHERE shelter_loc = 'Wisconsin'
      )
    )
)
```

Execution time: 0.047 seconds

	EMP_NAME
1	Jane Doe

3:20:56 AM - 1 rows total

4. List the shelters that have more employees than animals hosted in their locations:



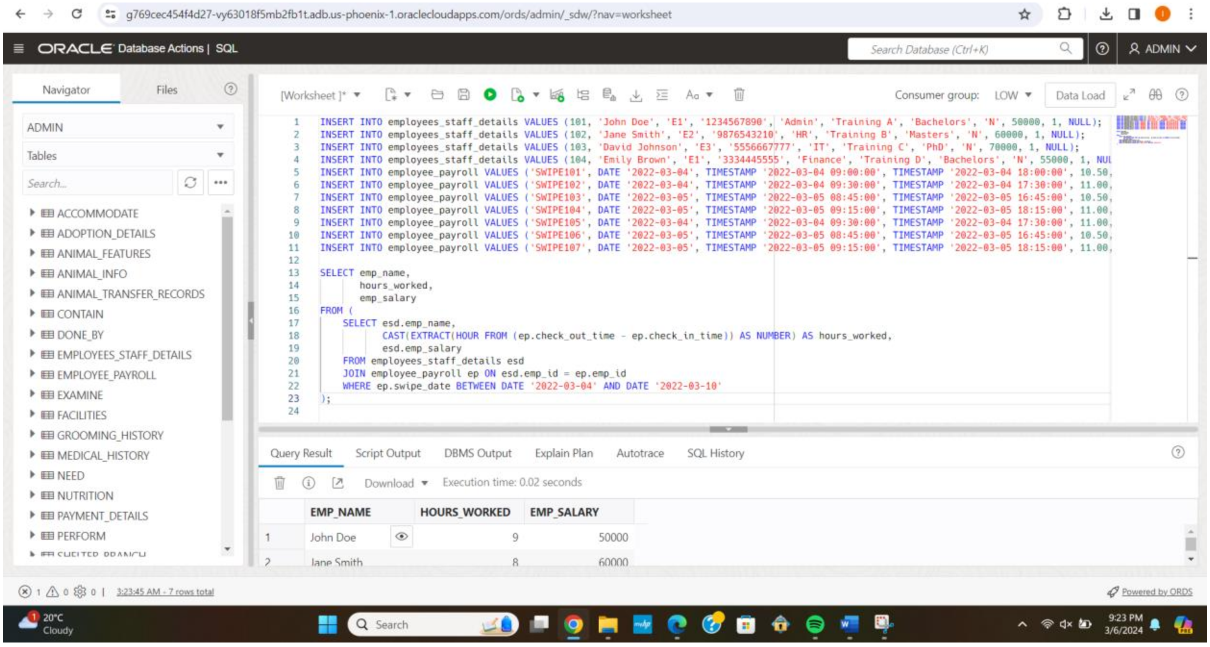
```
SELECT sb.shelter_loc
FROM shelter_branch sb
JOIN (
    SELECT shelter_id, COUNT(emp_id) AS employee_count
    FROM employees_staff_details
    GROUP BY shelter_id
) employees ON sb.shelter_id = employees.shelter_id
JOIN (
    SELECT shelter_id, COUNT(animal_id) AS animal_count
    FROM accommodate
    GROUP BY shelter_id
) animals ON sb.shelter_id = animals.shelter_id
WHERE employees.employee_count > animals.animal_count;
```

Execution time: 0.02 seconds

	SHELTER_LOC
1	Dallas
2	Wisconsin

3:22:39 AM - 2 rows total

5. Print the payroll from March 4, 2022 to March 10, 2022 displaying employee name, hours worked and total salary for all employees

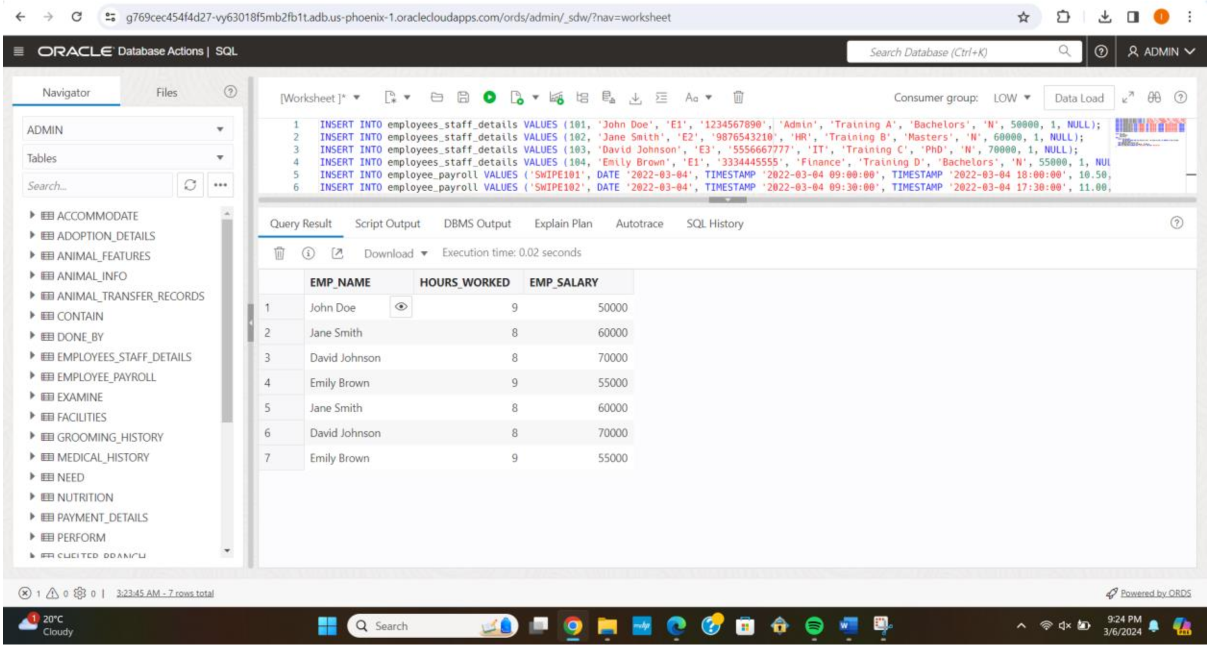


```
INSERT INTO employees_staff_details VALUES (101, 'John Doe', 'E1', '1234567890', 'Admin', 'Training A', 'Bachelors', 'N', 50000, 1, NULL);
INSERT INTO employees_staff_details VALUES (102, 'Jane Smith', 'E2', '9876543210', 'HR', 'Training B', 'Masters', 'N', 60000, 1, NULL);
INSERT INTO employees_staff_details VALUES (103, 'David Johnson', 'E3', '5556667777', 'IT', 'Training C', 'PhD', 'N', 70000, 1, NULL);
INSERT INTO employees_staff_details VALUES (104, 'Emily Brown', 'E1', '3334445555', 'Finance', 'Training D', 'Bachelors', 'N', 55000, 1, NUL
INSERT INTO employee_payroll VALUES ('SWIPE101', DATE '2022-03-04', TIMESTAMP '2022-03-04 09:00:00', TIMESTAMP '2022-03-04 18:00:00', 10.50,
INSERT INTO employee_payroll VALUES ('SWIPE102', DATE '2022-03-04', TIMESTAMP '2022-03-04 09:30:00', TIMESTAMP '2022-03-04 17:30:00', 11.00,
INSERT INTO employee_payroll VALUES ('SWIPE103', DATE '2022-03-05', TIMESTAMP '2022-03-05 08:45:00', TIMESTAMP '2022-03-05 16:45:00', 10.50,
INSERT INTO employee_payroll VALUES ('SWIPE104', DATE '2022-03-05', TIMESTAMP '2022-03-05 09:15:00', TIMESTAMP '2022-03-05 18:15:00', 11.00,
INSERT INTO employee_payroll VALUES ('SWIPE105', DATE '2022-03-04', TIMESTAMP '2022-03-04 09:30:00', TIMESTAMP '2022-03-04 17:30:00', 11.00,
INSERT INTO employee_payroll VALUES ('SWIPE106', DATE '2022-03-05', TIMESTAMP '2022-03-05 08:45:00', TIMESTAMP '2022-03-05 16:45:00', 10.50,
INSERT INTO employee_payroll VALUES ('SWIPE107', DATE '2022-03-05', TIMESTAMP '2022-03-05 09:15:00', TIMESTAMP '2022-03-05 18:15:00', 11.00,

SELECT emp_name,
       hours_worked,
       emp_salary
FROM (
    SELECT esd.emp_name,
           CAST(EXTRACT(HOUR FROM (ep.check_out_time - ep.check_in_time)) AS NUMBER) AS hours_worked,
           esd.emp_salary
    FROM employees_staff_details esd
    JOIN employee_payroll ep ON esd.emp_id = ep.emp_id
    WHERE ep.swipe_date BETWEEN DATE '2022-03-04' AND DATE '2022-03-10'
);
```

Execution time: 0.02 seconds

	EMP_NAME	HOURS_WORKED	EMP_SALARY
1	John Doe	9	50000
2	Jane Smith	8	60000

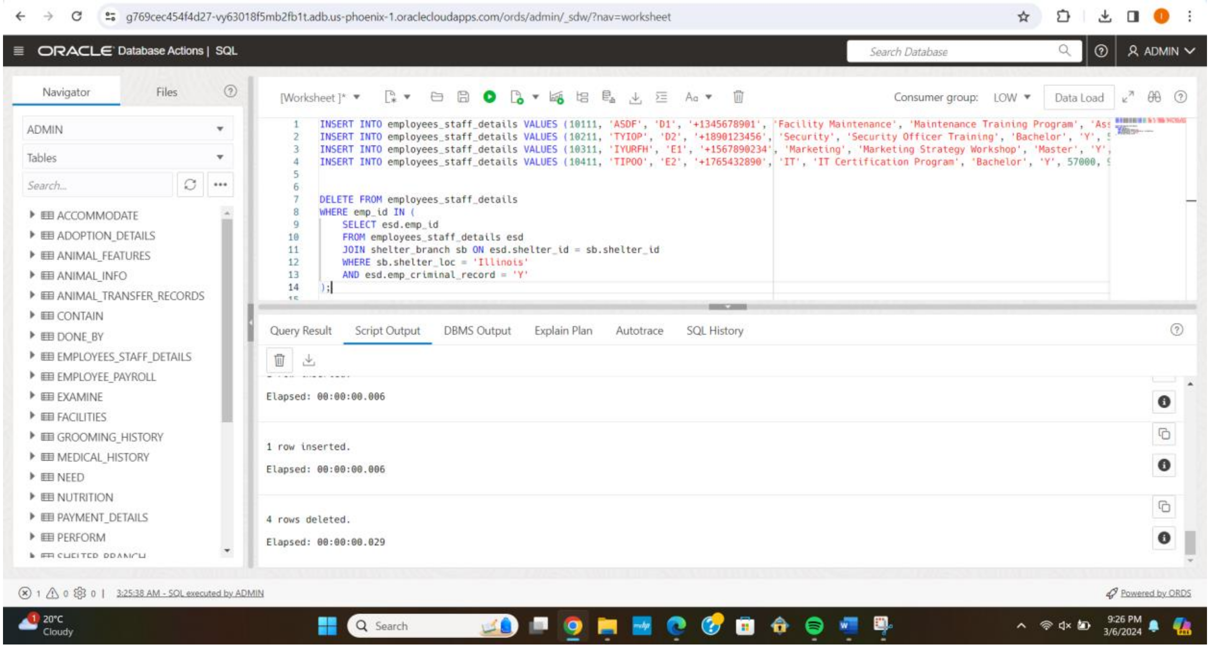
3:23:45 AM - 7 rows total



```sql
INSERT INTO employees_staff_details VALUES (101, 'John Doe', 'E1', '1234567890', 'Admin', 'Training A', 'Bachelors', 'N', 50000, 1, NULL);
INSERT INTO employees_staff_details VALUES (102, 'Jane Smith', 'E2', '9876543210', 'HR', 'Training B', 'Masters', 'N', 60000, 1, NULL);
INSERT INTO employees_staff_details VALUES (103, 'David Johnson', 'E3', '5556667777', 'IT', 'Training C', 'PhD', 'N', 70000, 1, NULL);
INSERT INTO employees_staff_details VALUES (104, 'Emily Brown', 'E1', '3334445555', 'Finance', 'Training D', 'Bachelors', 'N', 55000, 1, NUL
INSERT INTO employee_payroll VALUES ('SWIPE101', DATE '2022-03-04', TIMESTAMP '2022-03-04 09:00:00', TIMESTAMP '2022-03-04 18:00:00', 10.50,
INSERT INTO employee_payroll VALUES ('SWIPE102', DATE '2022-03-04', TIMESTAMP '2022-03-04 09:30:00', TIMESTAMP '2022-03-04 17:30:00', 11.00,
```

	EMP_NAME	HOURS_WORKED	EMP_SALARY
1	John Doe	9	50000
2	Jane Smith	8	60000
3	David Johnson	8	70000
4	Emily Brown	9	55000
5	Jane Smith	8	60000
6	David Johnson	8	70000
7	Emily Brown	9	55000

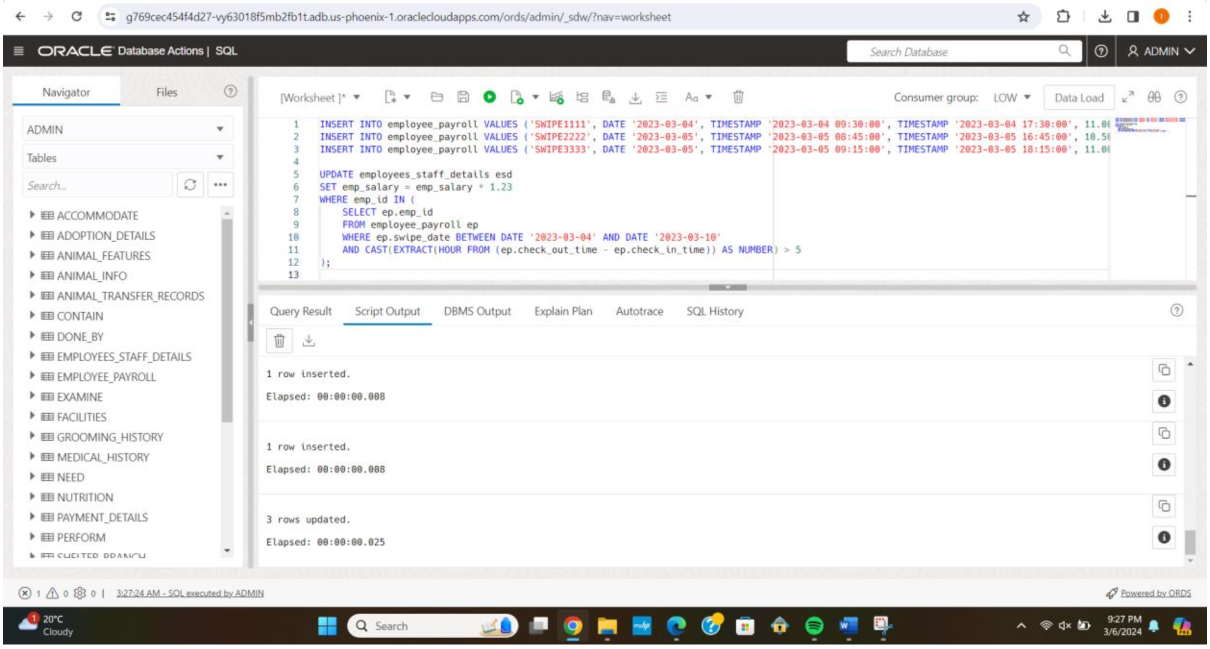
6. Design a delete statement to delete employees with criminal records working in Illinois locations:



```sql
INSERT INTO employees_staff_details VALUES (10111, 'ASDF', 'D1', '+1345678901', 'Facility Maintenance', 'Maintenance Training Program', 'Ass
INSERT INTO employees_staff_details VALUES (10211, 'TYIOP', 'D2', '+1890123456', 'Security', 'Security Officer Training', 'Bachelor', 'Y', 
INSERT INTO employees_staff_details VALUES (10311, 'IYURFH', 'E1', '+1567890234', 'Marketing', 'Marketing Strategy Workshop', 'Master', 'Y',
INSERT INTO employees_staff_details VALUES (10411, 'TIPOO', 'E2', '+1765432890', 'IT', 'IT Certification Program', 'Bachelor', 'Y', 57000, 


DELETE FROM employees_staff_details
WHERE emp_id IN (
    SELECT esd.emp_id
    FROM employees_staff_details esd
    JOIN shelter_branch sb ON esd.shelter_id = sb.shelter_id
    WHERE sb.shelter_loc = 'Illinois'
    AND esd.emp_criminal_record = 'Y'
);
```

```
Elapsed: 00:00:00.006


1 row inserted.

Elapsed: 00:00:00.006


4 rows deleted.

Elapsed: 00:00:00.029
```

7. Design an update statement to give a 23% salary raise to employees working more than 5 hours from March 4, 2023 to March 10, 2023:

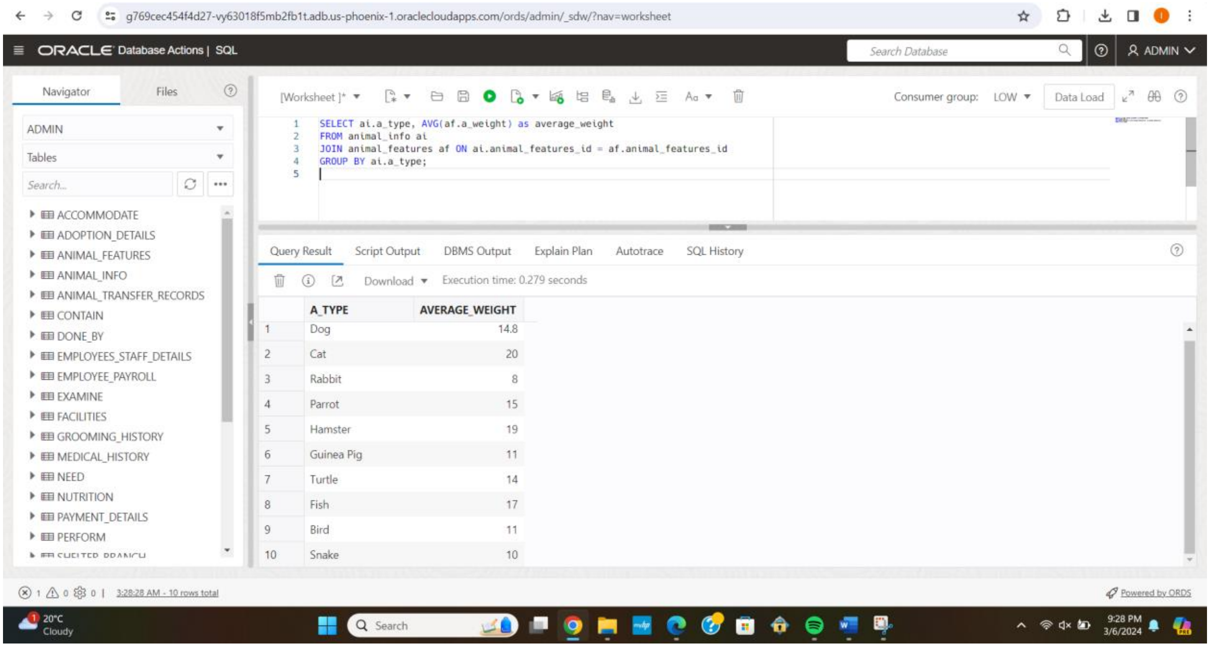


```sql
INSERT INTO employee_payroll VALUES ('SWIPE1111', DATE '2023-03-04', TIMESTAMP '2023-03-04 09:30:00', TIMESTAMP '2023-03-04 17:30:00', 11.0
INSERT INTO employee_payroll VALUES ('SWIPE2222', DATE '2023-03-05', TIMESTAMP '2023-03-05 08:45:00', TIMESTAMP '2023-03-05 16:45:00', 10.5
INSERT INTO employee_payroll VALUES ('SWIPE3333', DATE '2023-03-05', TIMESTAMP '2023-03-05 09:15:00', TIMESTAMP '2023-03-05 18:15:00', 11.0

UPDATE employees_staff_details esd
SET emp_salary = emp_salary * 1.23
WHERE emp_id IN (
    SELECT ep.emp_id
    FROM employee_payroll ep
    WHERE ep.swipe_date BETWEEN DATE '2023-03-04' AND DATE '2023-03-10'
    AND CAST(EXTRACT(HOUR FROM (ep.check_out_time - ep.check_in_time)) AS NUMBER) > 5
);
```

```
1 row inserted.

Elapsed: 00:00:00.008


1 row inserted.

Elapsed: 00:00:00.008


3 rows updated.

Elapsed: 00:00:00.025
```

Additional Queries:

1. Find Avg. Weight of animals per type?

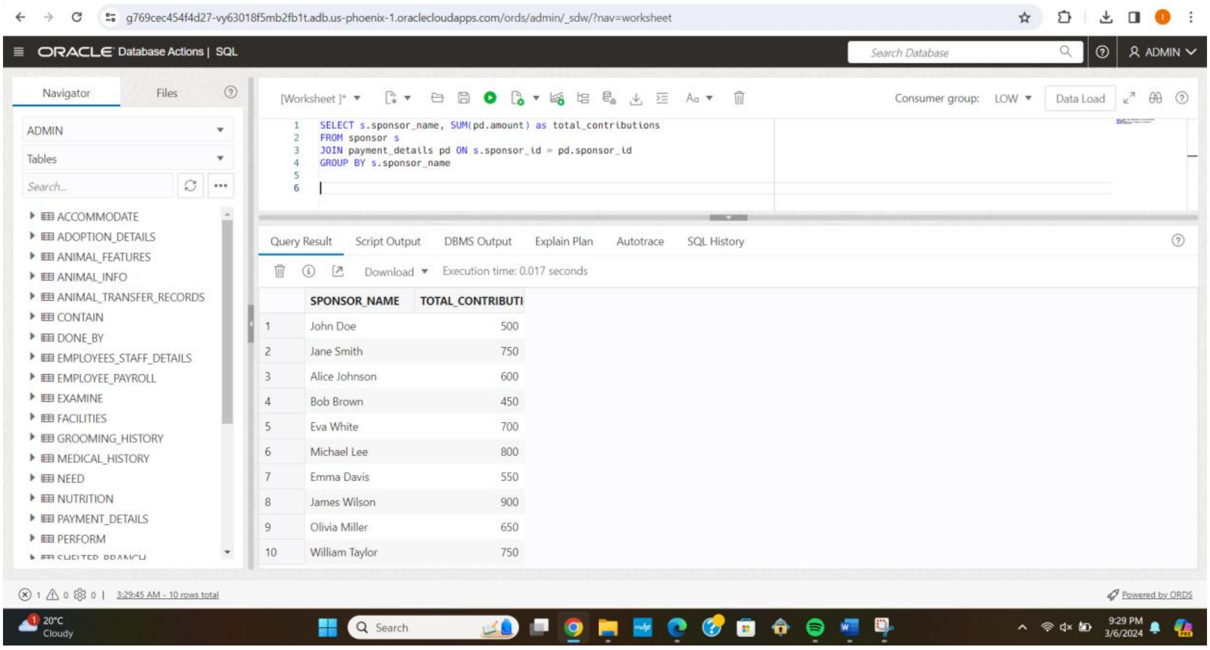


```sql
SELECT ai.a_type, AVG(af.a_weight) as average_weight
FROM animal_info ai
JOIN animal_features af ON ai.animal_features_id = af.animal_features_id
GROUP BY ai.a_type;
```

Execution time: 0.279 seconds

	A_TYPE	AVERAGE_WEIGHT
1	Dog	14.8
2	Cat	20
3	Rabbit	8
4	Parrot	15
5	Hamster	19
6	Guinea Pig	11
7	Turtle	14
8	Fish	17
9	Bird	11
10	Snake	10

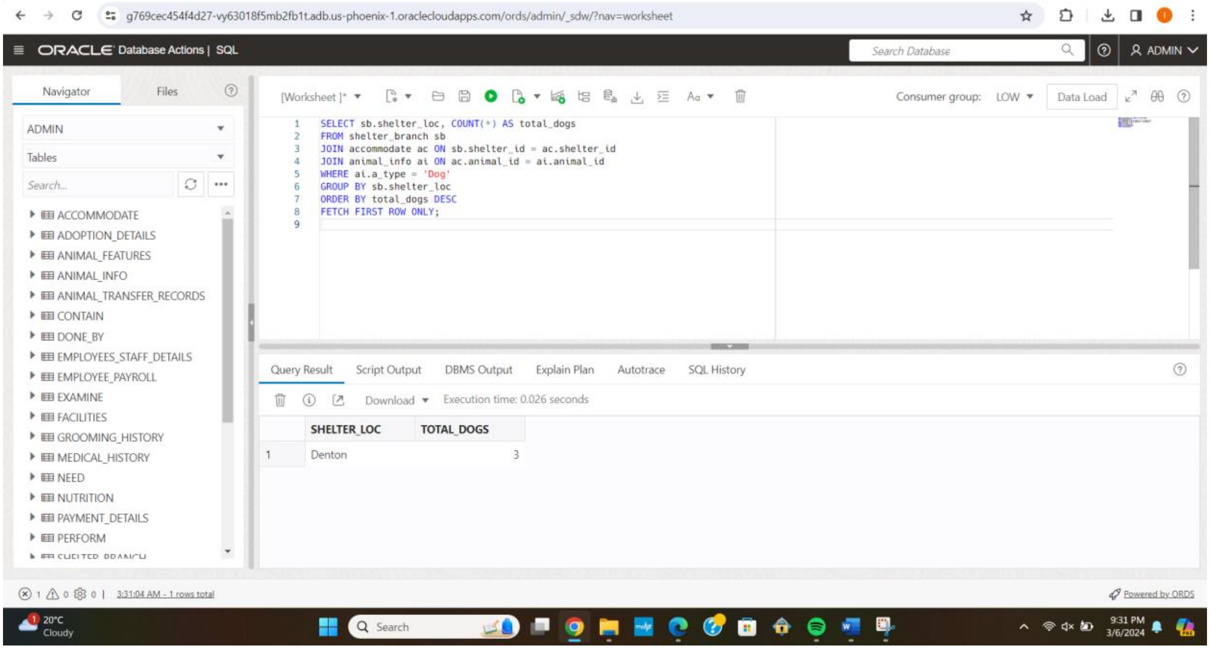
2. List the sponsors along with the total amount they have paid:



```sql
SELECT s.sponsor_name, SUM(pd.amount) as total_contributions
FROM sponsor s
JOIN payment_details pd ON s.sponsor_id = pd.sponsor_id
GROUP BY s.sponsor_name
```

	SPONSOR_NAME	TOTAL_CONTRIBUTI
1	John Doe	500
2	Jane Smith	750
3	Alice Johnson	600
4	Bob Brown	450
5	Eva White	700
6	Michael Lee	800
7	Emma Davis	550
8	James Wilson	900
9	Olivia Miller	650
10	William Taylor	750

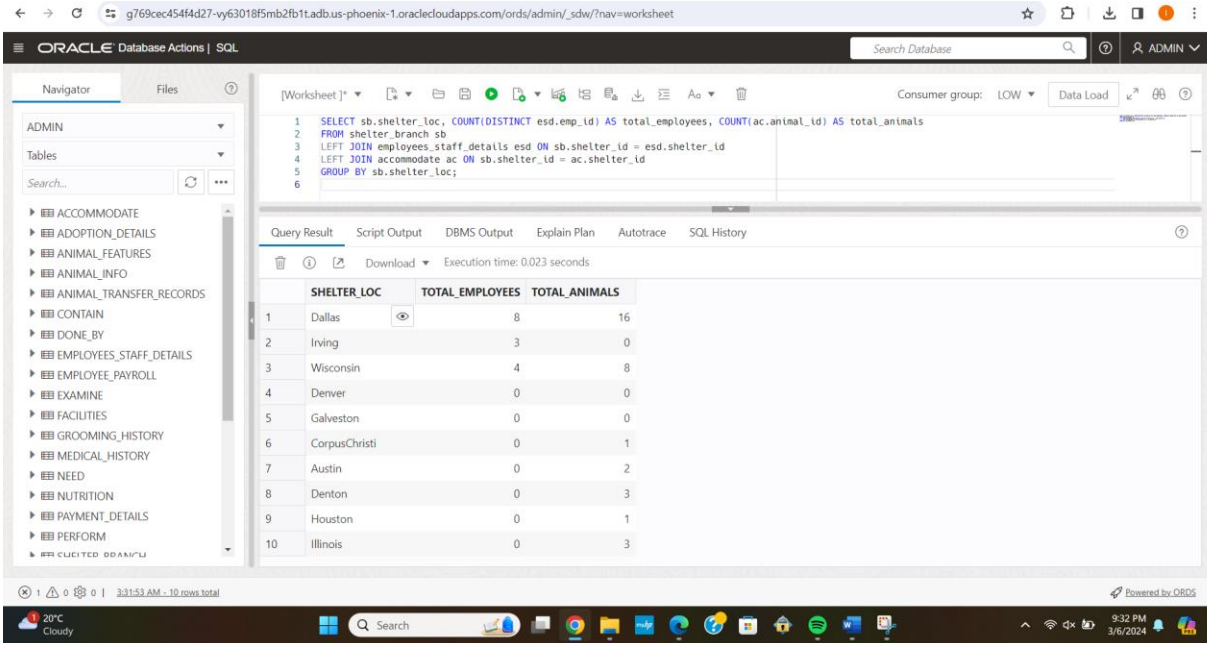
3. List the shelter locations with the highest number of dogs available for adoption:


g769cec454f4d27-vy63018f5mb2fb1t.adb.us-phoenix-1.oraclecloudapps.com/ords/admin/_sdw/?nav=worksheet
ORACLE Database Actions | SQL
Query Result
Download
Execution time: 0.026 seconds
3:31:04 AM - 1 rows total

```sql
SELECT sb.shelter_loc, COUNT(*) AS total_dogs
FROM shelter_branch sb
JOIN accommodate ac ON sb.shelter_id = ac.shelter_id
JOIN animal_info ai ON ac.animal_id = ai.animal_id
WHERE ai.a_type = 'Dog'
GROUP BY sb.shelter_loc
ORDER BY total_dogs DESC
FETCH FIRST ROW ONLY;
```

	SHELTER_LOC	TOTAL_DOGS
1	Denton	3

4. Find the total number of employees and animals hosted in each shelter location:



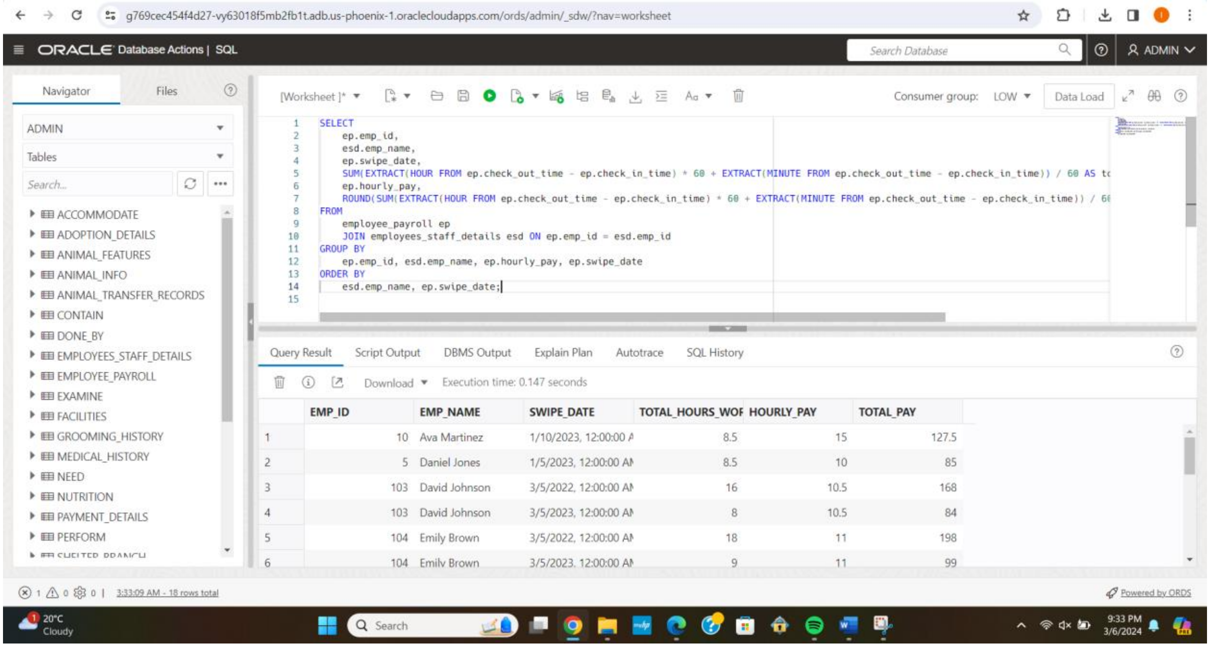
```sql
SELECT sb.shelter_loc, COUNT(DISTINCT esd.emp_id) AS total_employees, COUNT(ac.animal_id) AS total_animals
FROM shelter_branch sb
LEFT JOIN employees_staff_details esd ON sb.shelter_id = esd.shelter_id
LEFT JOIN accommodate ac ON sb.shelter_id = ac.shelter_id
GROUP BY sb.shelter_loc;
```

Execution time: 0.023 seconds

	SHELTER_LOC	TOTAL_EMPLOYEES	TOTAL_ANIMALS
1	Dallas	8	16
2	Irving	3	0
3	Wisconsin	4	8
4	Denver	0	0
5	Galveston	0	0
6	CorpusChristi	0	1
7	Austin	0	2
8	Denton	0	3
9	Houston	0	1
10	Illinois	0	3

3:31:53 AM - 10 rows total

5. Calculate the total hours worked and total pay for each employee:

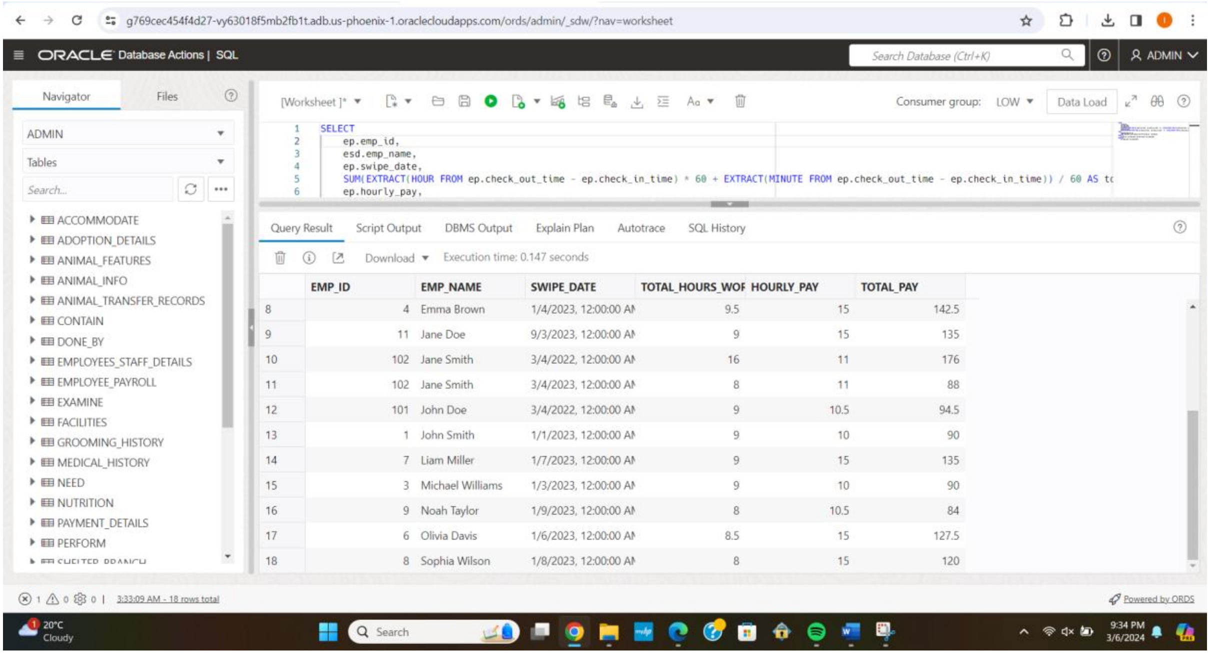


```sql
SELECT
    ep.emp_id,
    esd.emp_name,
    ep.swipe_date,
    SUM(EXTRACT(HOUR FROM ep.check_out_time - ep.check_in_time) * 60 + EXTRACT(MINUTE FROM ep.check_out_time - ep.check_in_time)) / 60 AS tc
    ep.hourly_pay,
    ROUND(SUM(EXTRACT(HOUR FROM ep.check_out_time - ep.check_in_time) * 60 + EXTRACT(MINUTE FROM ep.check_out_time - ep.check_in_time)) / 6(
FROM
    employee_payroll ep
    JOIN employees_staff_details esd ON ep.emp_id = esd.emp_id
GROUP BY
    ep.emp_id, esd.emp_name, ep.hourly_pay, ep.swipe_date
ORDER BY
    esd.emp_name, ep.swipe_date;
```

Execution time: 0.147 seconds

	EMP_ID	EMP_NAME	SWIPE_DATE	TOTAL_HOURS_WOF	HOURLY_PAY	TOTAL_PAY
1	10	Ava Martinez	1/10/2023, 12:00:00 A	8.5	15	127.5
2	5	Daniel Jones	1/5/2023, 12:00:00 AI	8.5	10	85
3	103	David Johnson	3/5/2022, 12:00:00 AI	16	10.5	168
4	103	David Johnson	3/5/2023, 12:00:00 AI	8	10.5	84
5	104	Emily Brown	3/5/2022, 12:00:00 AI	18	11	198
6	104	Emily Brown	3/5/2023, 12:00:00 AI	9	11	99

3:33:09 AM - 18 rows total

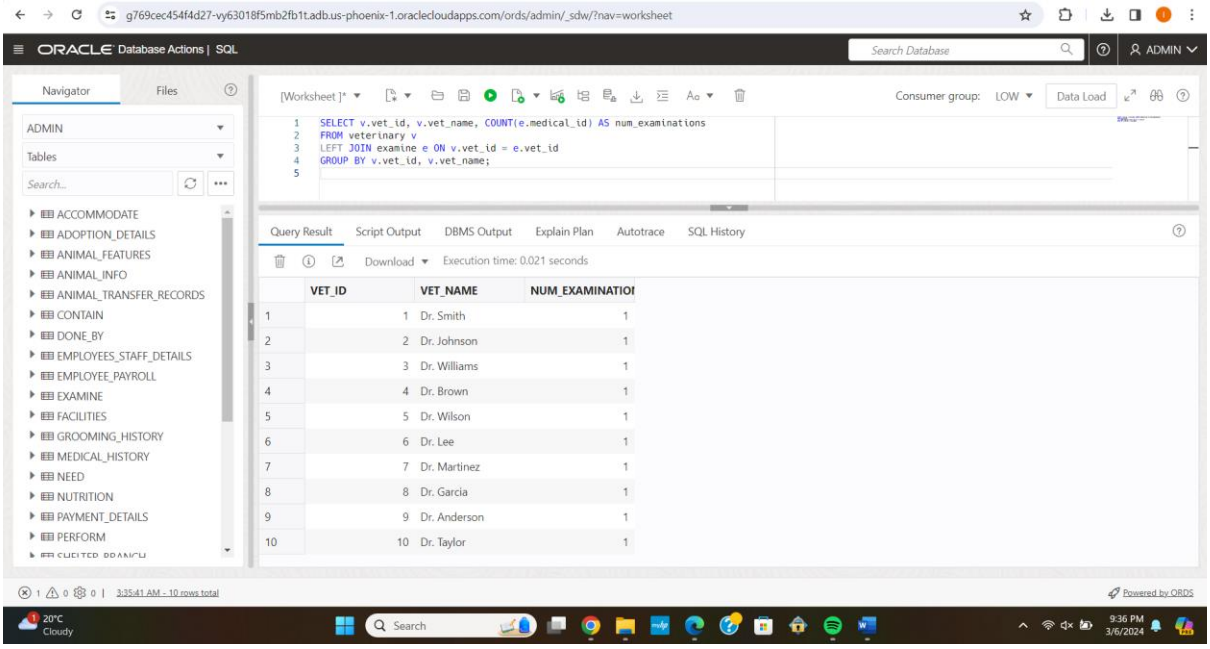


```
SELECT
    ep.emp_id,
    esd.emp_name,
    ep.swipe_date,
    SUM(EXTRACT(HOUR FROM ep.check_out_time - ep.check_in_time) * 60 + EXTRACT(MINUTE FROM ep.check_out_time - ep.check_in_time)) / 60 AS tc
    ep.hourly_pay,
```

Execution time: 0.147 seconds

	EMP_ID	EMP_NAME	SWIPE_DATE	TOTAL_HOURS_WOF	HOURLY_PAY	TOTAL_PAY
8	4	Emma Brown	1/4/2023, 12:00:00 AI	9.5	15	142.5
9	11	Jane Doe	9/3/2023, 12:00:00 AI	9	15	135
10	102	Jane Smith	3/4/2022, 12:00:00 AI	16	11	176
11	102	Jane Smith	3/4/2023, 12:00:00 AI	8	11	88
12	101	John Doe	3/4/2022, 12:00:00 AI	9	10.5	94.5
13	1	John Smith	1/1/2023, 12:00:00 AI	9	10	90
14	7	Liam Miller	1/7/2023, 12:00:00 AI	9	15	135
15	3	Michael Williams	1/3/2023, 12:00:00 AI	9	10	90
16	9	Noah Taylor	1/9/2023, 12:00:00 AI	8	10.5	84
17	6	Olivia Davis	1/6/2023, 12:00:00 AI	8.5	15	127.5
18	8	Sophia Wilson	1/8/2023, 12:00:00 AI	8	15	120

6. Find the number of examinations conducted by each veterinarian:

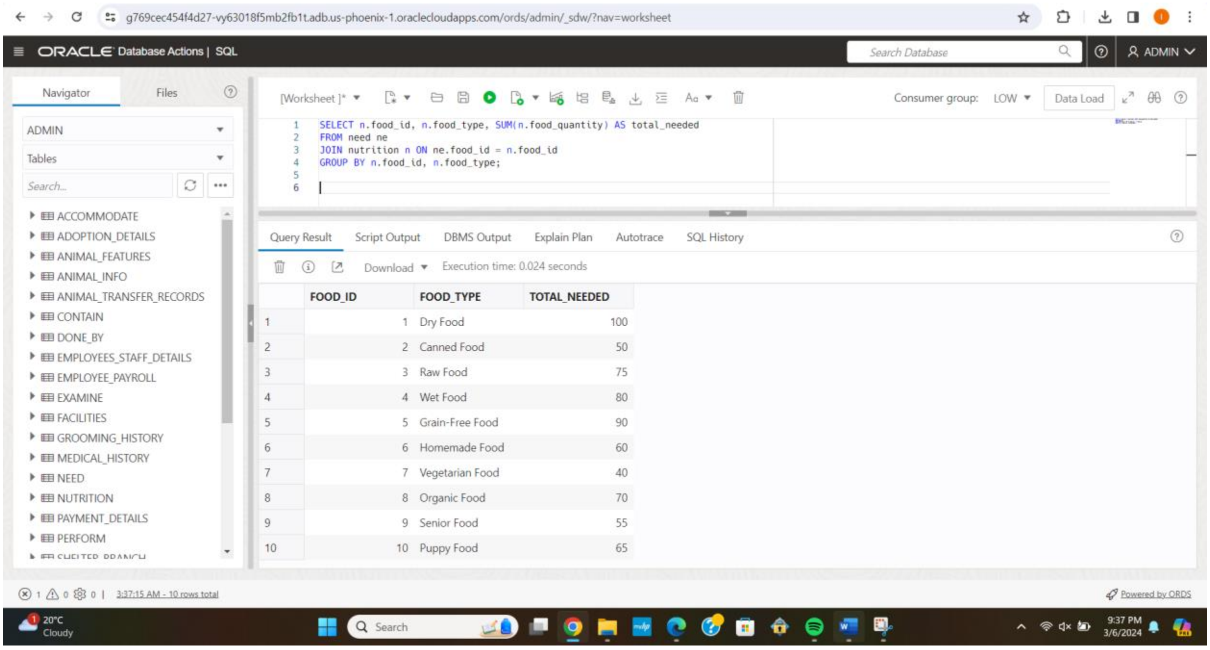


```
SELECT v.vet_id, v.vet_name, COUNT(e.medical_id) AS num_examinations
FROM veterinary v
LEFT JOIN examine e ON v.vet_id = e.vet_id
GROUP BY v.vet_id, v.vet_name;
```

Execution time: 0.021 seconds

	VET_ID	VET_NAME	NUM_EXAMINATIOI
1	1	Dr. Smith	1
2	2	Dr. Johnson	1
3	3	Dr. Williams	1
4	4	Dr. Brown	1
5	5	Dr. Wilson	1
6	6	Dr. Lee	1
7	7	Dr. Martinez	1
8	8	Dr. Garcia	1
9	9	Dr. Anderson	1
10	10	Dr. Taylor	1

7. Find the total quantity of each type of food needed for all animals:



g769cec454f4d27-vy63018f5mb2fb1t.adb.us-phoenix-1.oraclecloudapps.com/ords/admin/_sdw/?nav=worksheet

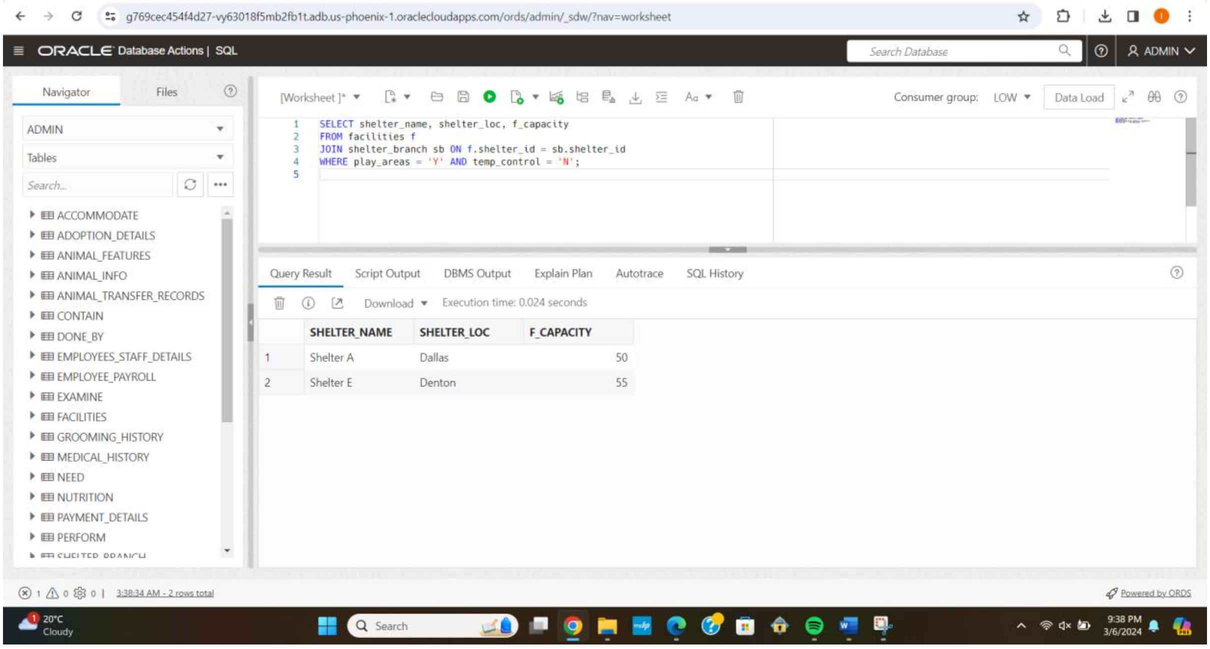
```sql
SELECT n.food_id, n.food_type, SUM(n.food_quantity) AS total_needed
FROM need ne
JOIN nutrition n ON ne.food_id = n.food_id
GROUP BY n.food_id, n.food_type;
```

Execution time: 0.024 seconds

	FOOD_ID	FOOD_TYPE	TOTAL_NEEDED
1	1	Dry Food	100
2	2	Canned Food	50
3	3	Raw Food	75
4	4	Wet Food	80
5	5	Grain-Free Food	90
6	6	Homemade Food	60
7	7	Vegetarian Food	40
8	8	Organic Food	70
9	9	Senior Food	55
10	10	Puppy Food	65

3:37:15 AM - 10 rows total

8. List of All Facilities with Play Areas but No Temperature Control



g769cec454f4d27-vy63018f5mb2fb1t.adb.us-phoenix-1.oraclecloudapps.com/ords/admin/_sdw/?nav=worksheet

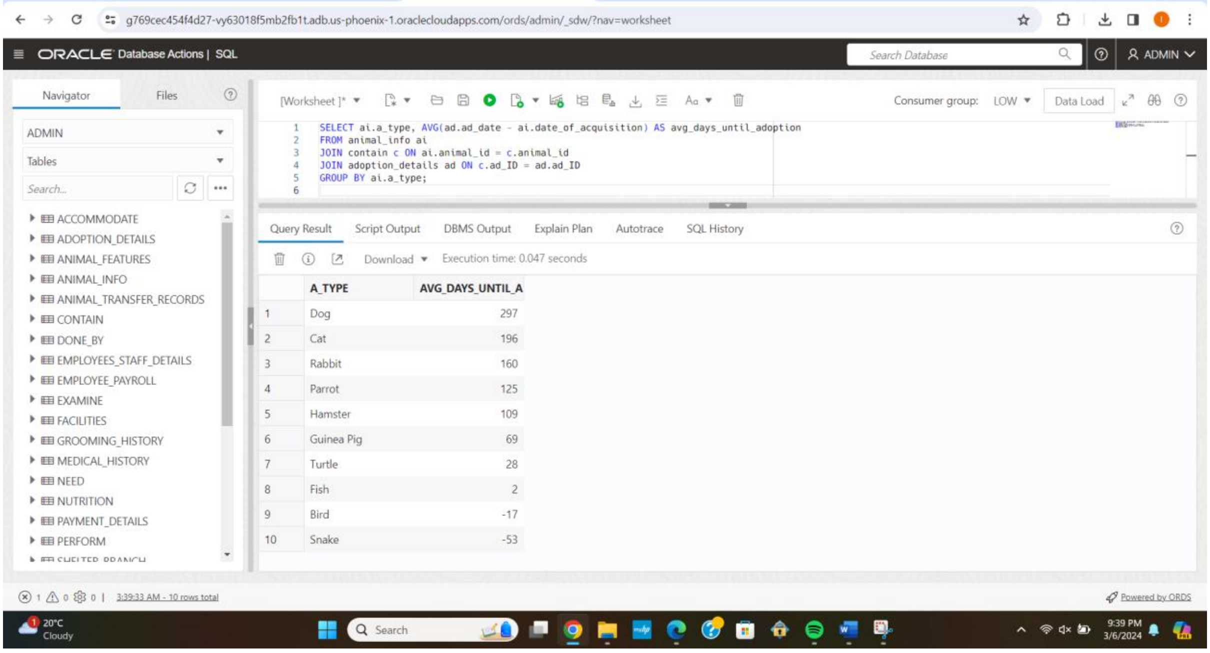
```sql
SELECT shelter_name, shelter_loc, f_capacity
FROM facilities f
JOIN shelter_branch sb ON f.shelter_id = sb.shelter_id
WHERE play_areas = 'Y' AND temp_control = 'N';
```

Execution time: 0.024 seconds

	SHELTER_NAME	SHELTER_LOC	F_CAPACITY
1	Shelter A	Dallas	50
2	Shelter E	Denton	55

3:38:34 AM - 2 rows total

9. Average Duration Between Animal Acquisition and Adoption



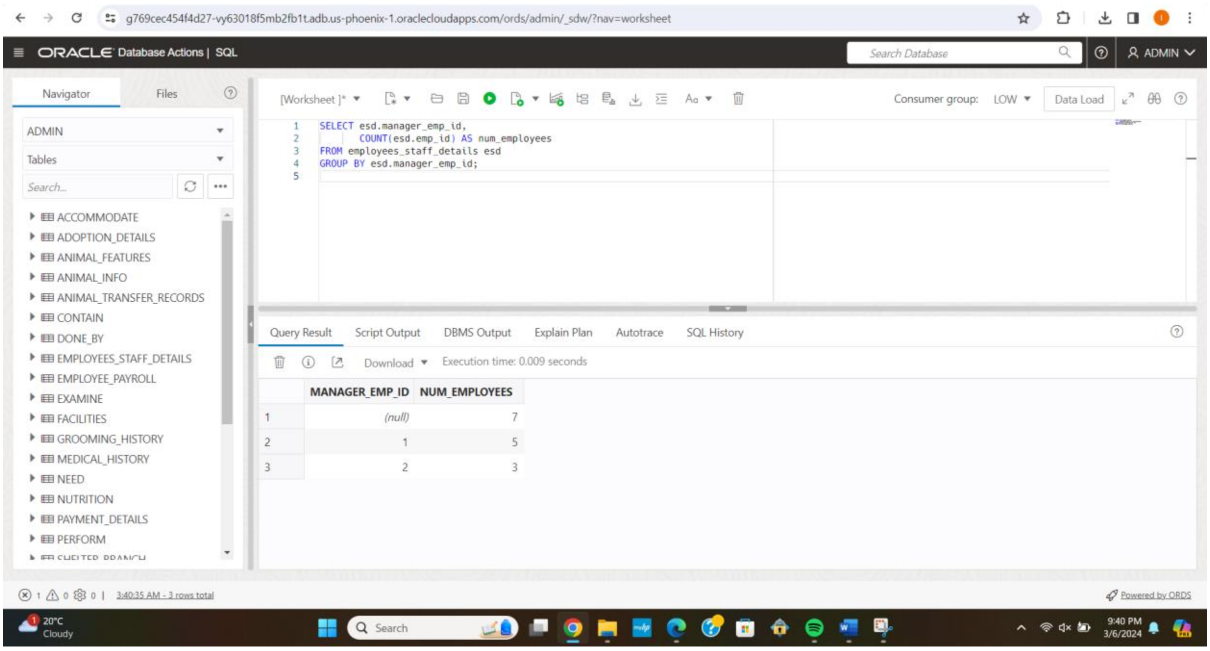
```sql
SELECT ai.a_type, AVG(ad.ad_date - ai.date_of_acquisition) AS avg_days_until_adoption
FROM animal_info ai
JOIN contain c ON ai.animal_id = c.animal_id
JOIN adoption_details ad ON c.ad_ID = ad.ad_ID
GROUP BY ai.a_type;
```

Execution time: 0.047 seconds

	A_TYPE	AVG_DAYS_UNTIL_A
1	Dog	297
2	Cat	196
3	Rabbit	160
4	Parrot	125
5	Hamster	109
6	Guinea Pig	69
7	Turtle	28
8	Fish	2
9	Bird	-17
10	Snake	-53

3:39:33 AM - 10 rows total

10. Find the number of employees under each manager:



```sql
SELECT esd.manager_emp_id,
       COUNT(esd.emp_id) AS num_employees
FROM employees_staff_details esd
GROUP BY esd.manager_emp_id;
```

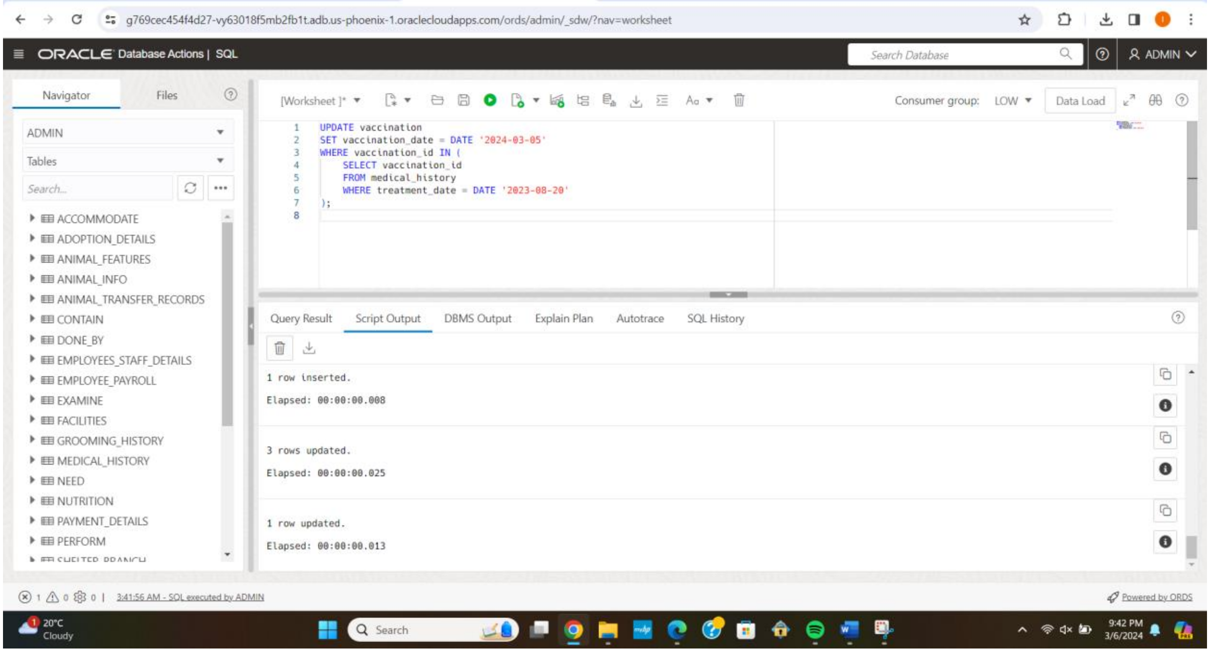
Execution time: 0.009 seconds

	MANAGER_EMP_ID	NUM_EMPLOYEES
1	(null)	7
2	1	5
3	2	3

3:40:35 AM - 3 rows total

Update Statements:

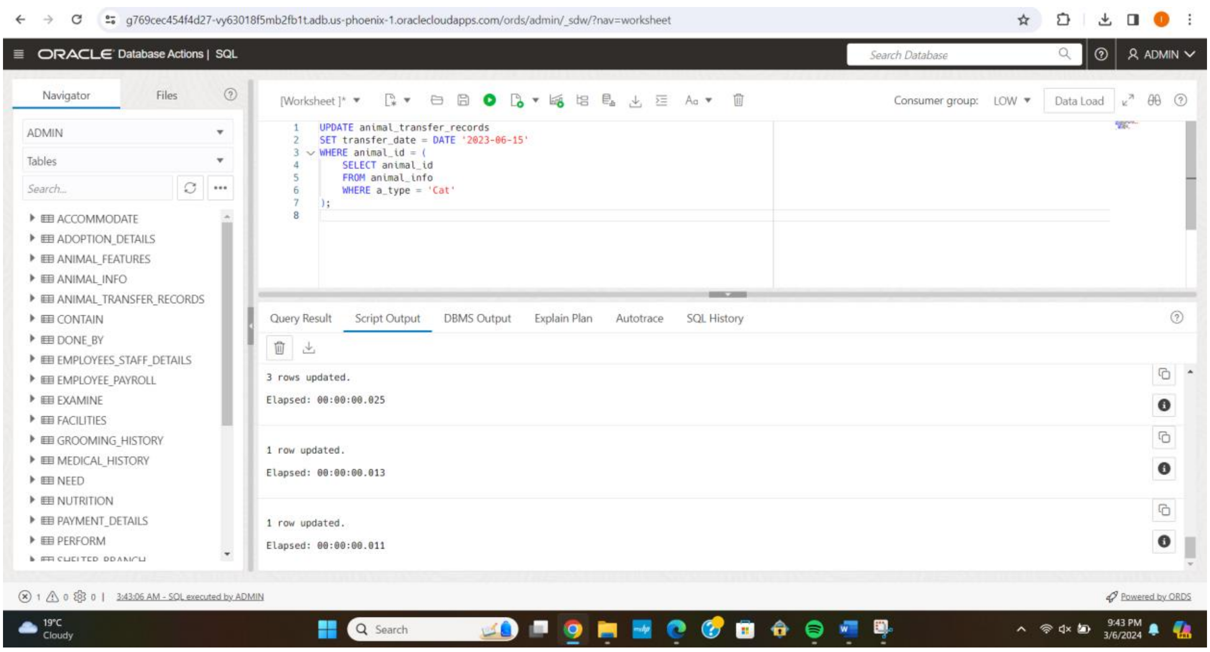
1. Update the vaccination date when the treatment_date is : 2023-08-20



```sql
UPDATE vaccination
SET vaccination_date = DATE '2024-03-05'
WHERE vaccination_id IN (
    SELECT vaccination_id
    FROM medical_history
    WHERE treatment_date = DATE '2023-08-20'
);
```

```
1 row inserted.

Elapsed: 00:00:00.008


3 rows updated.

Elapsed: 00:00:00.025


1 row updated.

Elapsed: 00:00:00.013
```

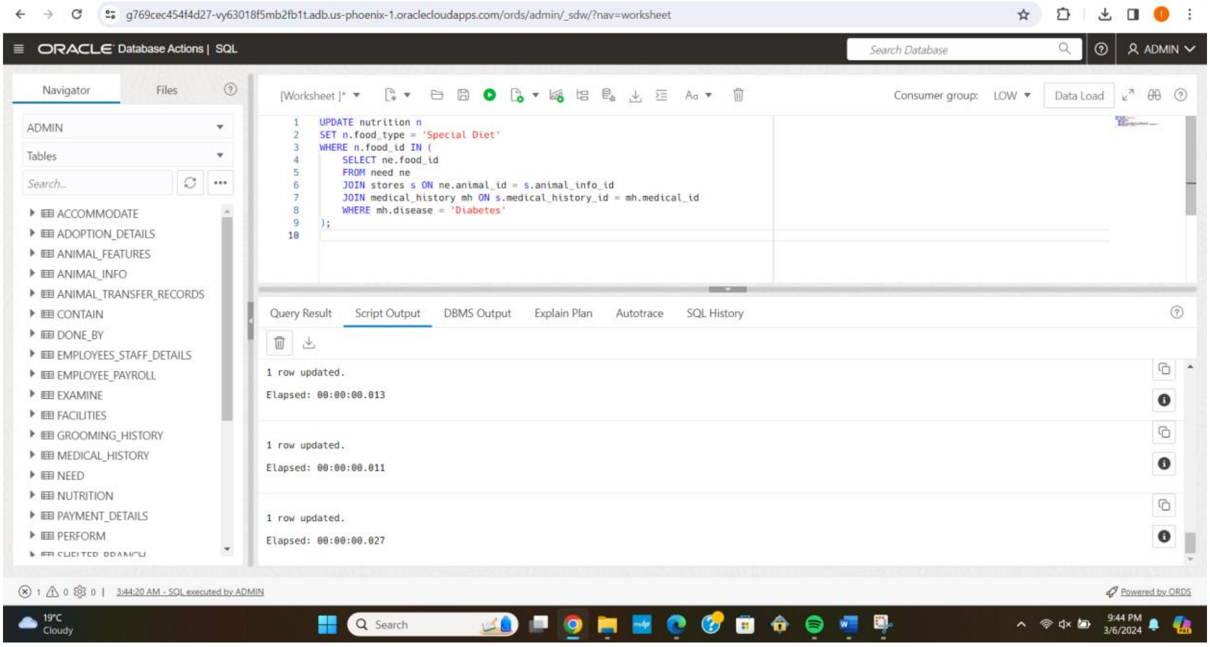
2. Update the transfer date for a specific animal:



```sql
UPDATE animal_transfer_records
SET transfer_date = DATE '2023-06-15'
WHERE animal_id = (
    SELECT animal_id
    FROM animal_info
    WHERE a_type = 'Cat'
);
```

```
3 rows updated.

Elapsed: 00:00:00.025


1 row updated.

Elapsed: 00:00:00.013


1 row updated.

Elapsed: 00:00:00.011
```

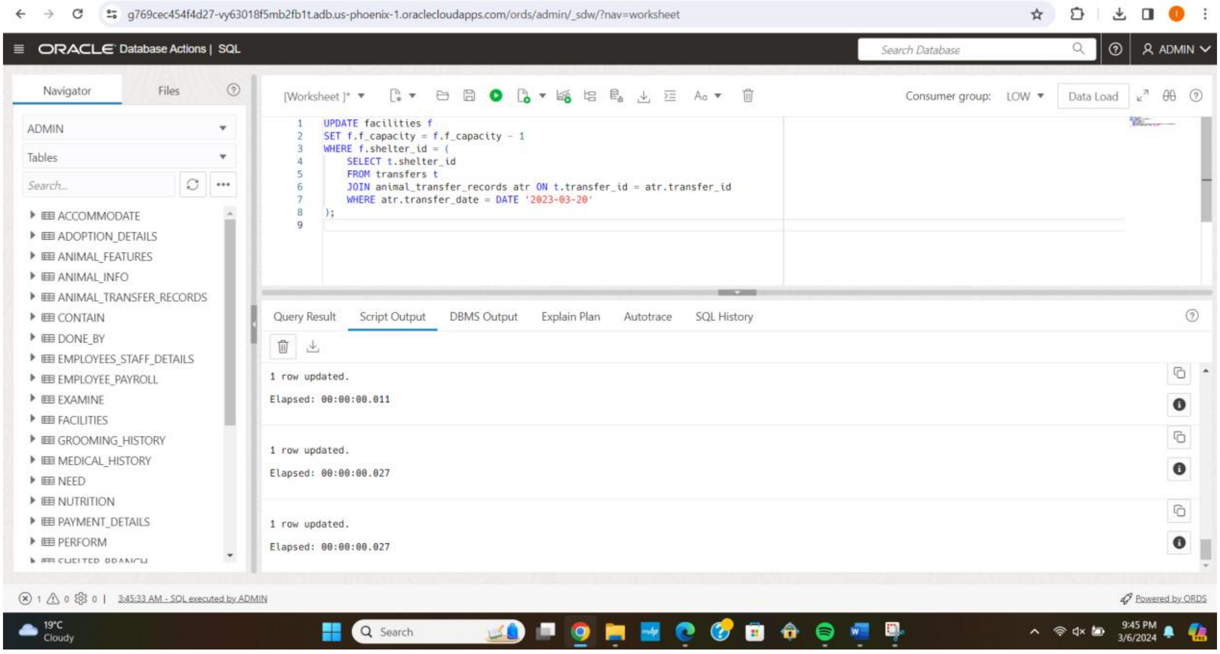
3. Update the food type for animals with diabetes in the nutrition table



```sql
UPDATE nutrition n
SET n.food_type = 'Special Diet'
WHERE n.food_id IN (
    SELECT ne.food_id
    FROM need ne
    JOIN stores s ON ne.animal_id = s.animal_info_id
    JOIN medical_history mh ON s.medical_history_id = mh.medical_id
    WHERE mh.disease = 'Diabetes'
);
```

```
1 row updated.

Elapsed: 00:00:00.013


1 row updated.

Elapsed: 00:00:00.011


1 row updated.

Elapsed: 00:00:00.027
```

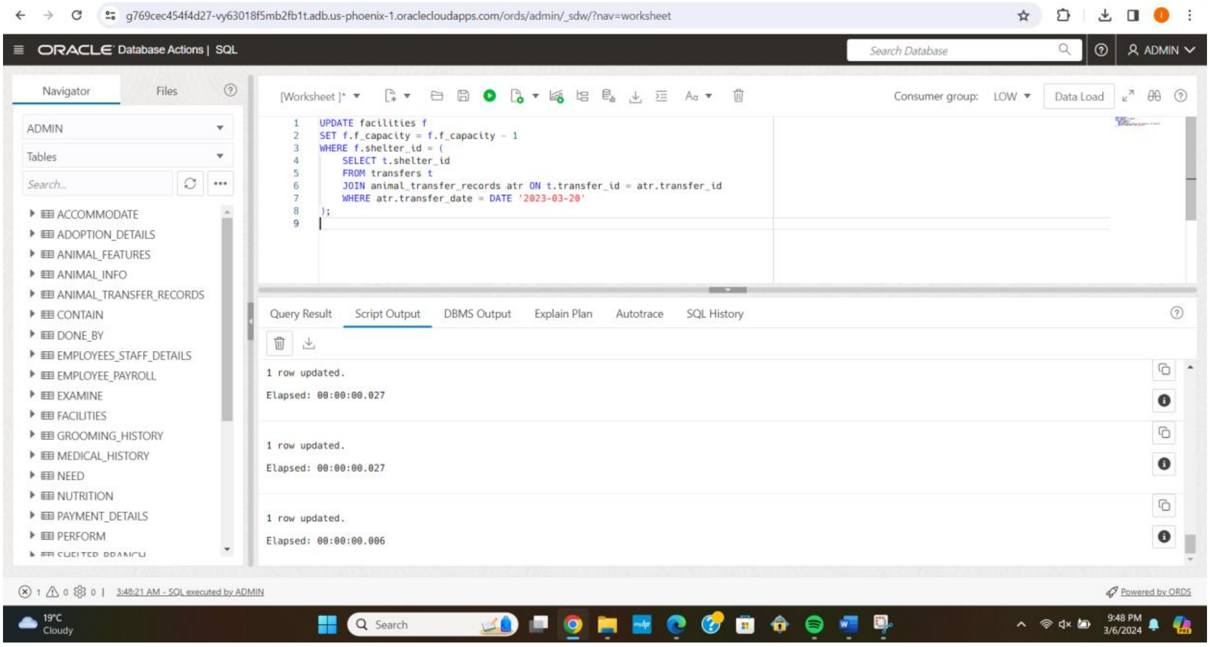
4. Update shelter's capacity based on the number of animals transferred out



```sql
UPDATE facilities f
SET f.f_capacity = f.f_capacity - 1
WHERE f.shelter_id = (
    SELECT t.shelter_id
    FROM transfers t
    JOIN animal_transfer_records atr ON t.transfer_id = atr.transfer_id
    WHERE atr.transfer_date = DATE '2023-03-20'
);
```

```
1 row updated.

Elapsed: 00:00:00.011


1 row updated.

Elapsed: 00:00:00.027


1 row updated.

Elapsed: 00:00:00.027
```

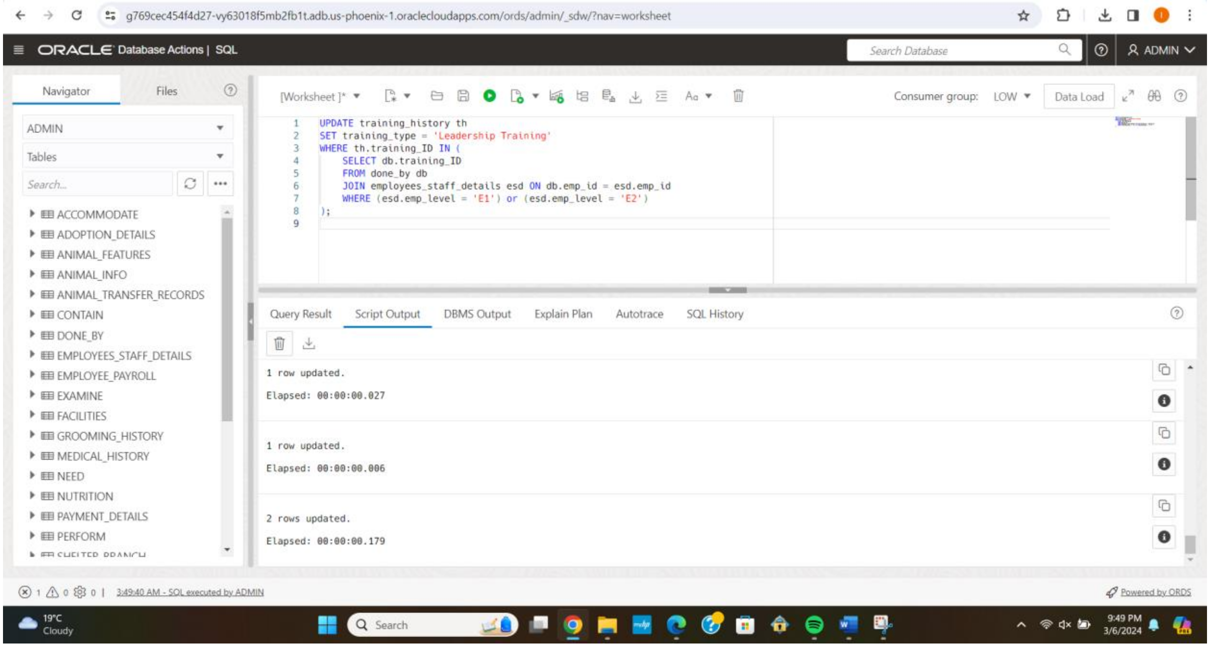
5. Update grooming type for animals with certain features:



```sql
UPDATE facilities f
SET f.f_capacity = f.f_capacity - 1
WHERE f.shelter_id = (
    SELECT t.shelter_id
    FROM transfers t
    JOIN animal_transfer_records atr ON t.transfer_id = atr.transfer_id
    WHERE atr.transfer_date = DATE '2023-03-20'
);
```

```
1 row updated.

Elapsed: 00:00:00.027


1 row updated.

Elapsed: 00:00:00.027


1 row updated.

Elapsed: 00:00:00.006
```

6. Update the training type for employees at top level

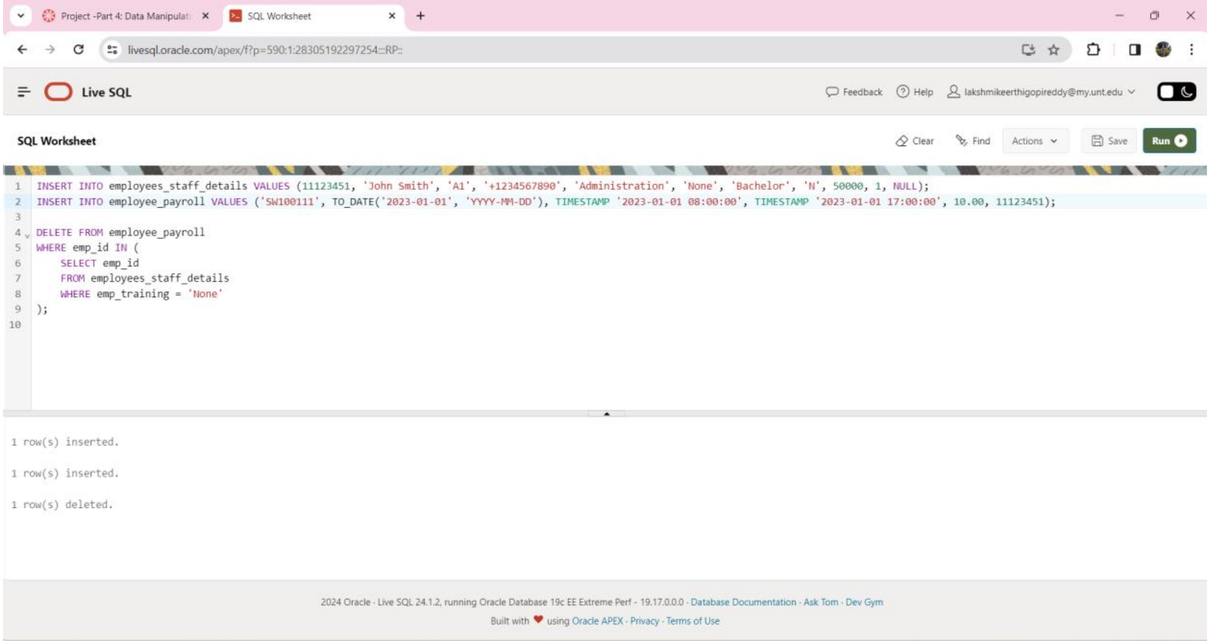


```sql
UPDATE training_history th
SET training_type = 'Leadership Training'
WHERE th.training_ID IN (
    SELECT db.training_ID
    FROM done_by db
    JOIN employees_staff_details esd ON db.emp_id = esd.emp_id
    WHERE (esd.emp_level = 'E1') or (esd.emp_level = 'E2')
);
```

```
1 row updated.

Elapsed: 00:00:00.027


1 row updated.

Elapsed: 00:00:00.006


2 rows updated.

Elapsed: 00:00:00.179
```

Delete Statements:

1. Delete the payroll of the employees who have not done any training:



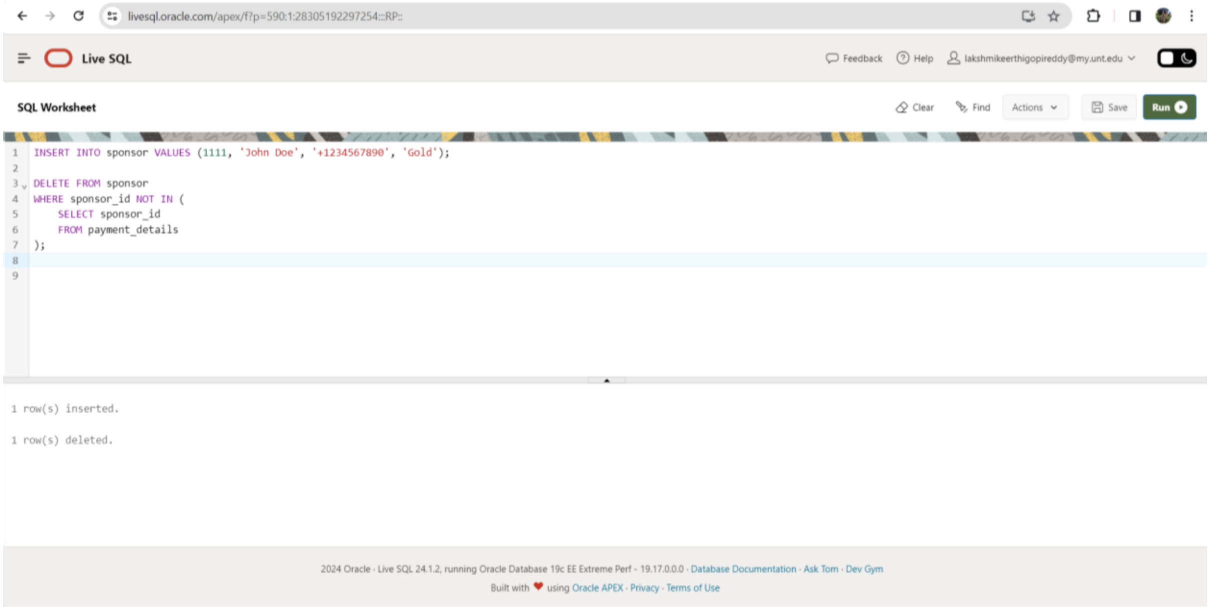
Project -Part 4: Data Manipulati
SQL Worksheet
livesql.oracle.com/apex/f?p=590:1:28305192297254::RP::
Live SQL
Feedback
Help
lakshmikeerthigopireddy@my.unt.edu
SQL Worksheet
Clear
Find
Actions
Save
Run

```sql
INSERT INTO employees_staff_details VALUES (11123451, 'John Smith', 'A1', '+1234567890', 'Administration', 'None', 'Bachelor', 'N', 50000, 1, NULL);
INSERT INTO employee_payroll VALUES ('SW100111', TO_DATE('2023-01-01', 'YYYY-MM-DD'), TIMESTAMP '2023-01-01 08:00:00', TIMESTAMP '2023-01-01 17:00:00', 10.00, 11123451);

DELETE FROM employee_payroll
WHERE emp_id IN (
    SELECT emp_id
    FROM employees_staff_details
    WHERE emp_training = 'None'
);
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) deleted.
```

2024 Oracle · Live SQL 24.1.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

2. Delete all records from the sponsor table for sponsors who have not made any payments:



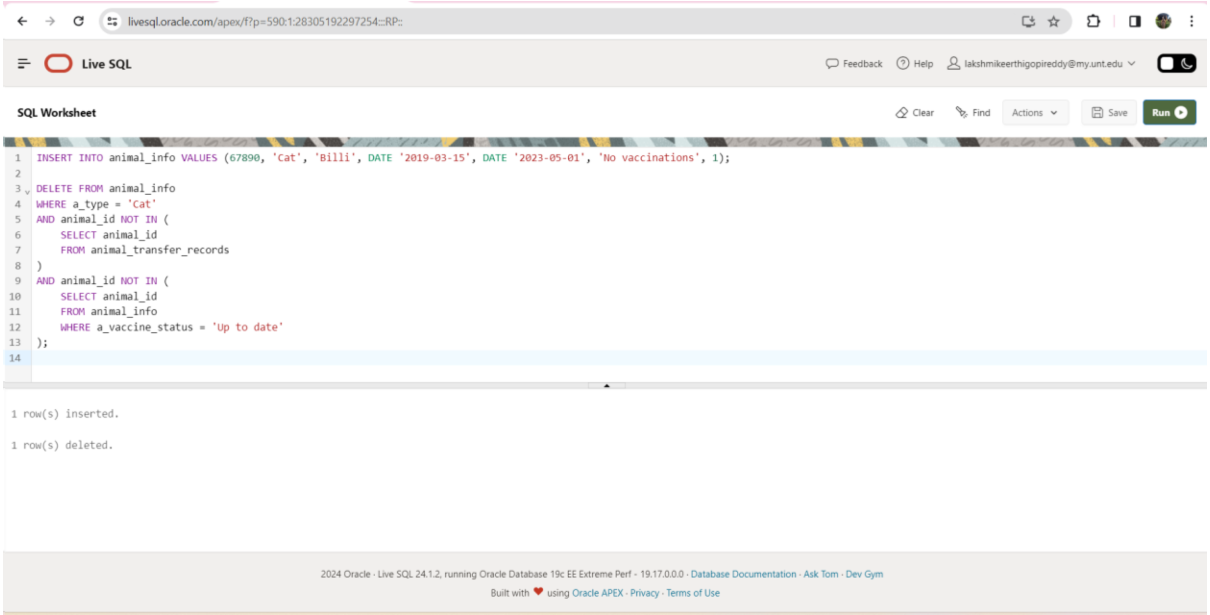
livesql.oracle.com/apex/f?p=590:1:28305192297254::RP::
Live SQL
Feedback
Help
lakshmikeerthigopireddy@my.unt.edu
SQL Worksheet
Clear
Find
Actions
Save
Run

```sql
INSERT INTO sponsor VALUES (1111, 'John Doe', '+1234567890', 'Gold');

DELETE FROM sponsor
WHERE sponsor_id NOT IN (
    SELECT sponsor_id
    FROM payment_details
);
```

```
1 row(s) inserted.

1 row(s) deleted.
```

2024 Oracle · Live SQL 24.1.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ♥ using Oracle APEX · Privacy · Terms of Use

3. Delete all records from the animal_info table where the animals are of type 'Cat', and they have not been transferred to any other shelter, and their vaccination status is not up to date

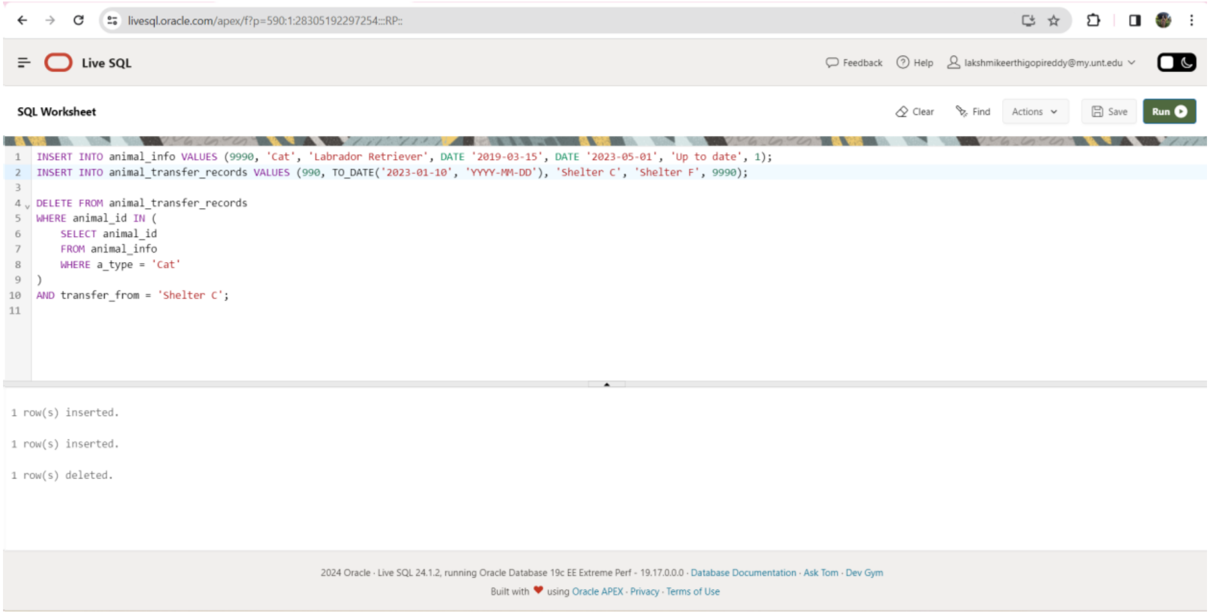

livesql.oracle.com/apex/f?p=590:1:28305192297254::RP::
Live SQL
Feedback
Help
lakshmikeerthigopireddy@my.unt.edu
SQL Worksheet
Clear
Find
Actions
Save
Run

```sql
INSERT INTO animal_info VALUES (67890, 'Cat', 'Billi', DATE '2019-03-15', DATE '2023-05-01', 'No vaccinations', 1);

DELETE FROM animal_info
WHERE a_type = 'Cat'
AND animal_id NOT IN (
    SELECT animal_id
    FROM animal_transfer_records
)
AND animal_id NOT IN (
    SELECT animal_id
    FROM animal_info
    WHERE a_vaccine_status = 'Up to date'
);
```

```
1 row(s) inserted.

1 row(s) deleted.
```

2024 Oracle · Live SQL 24.1.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

4. Delete the animal transfer record if the animal type is cat and the location is Shelter C:

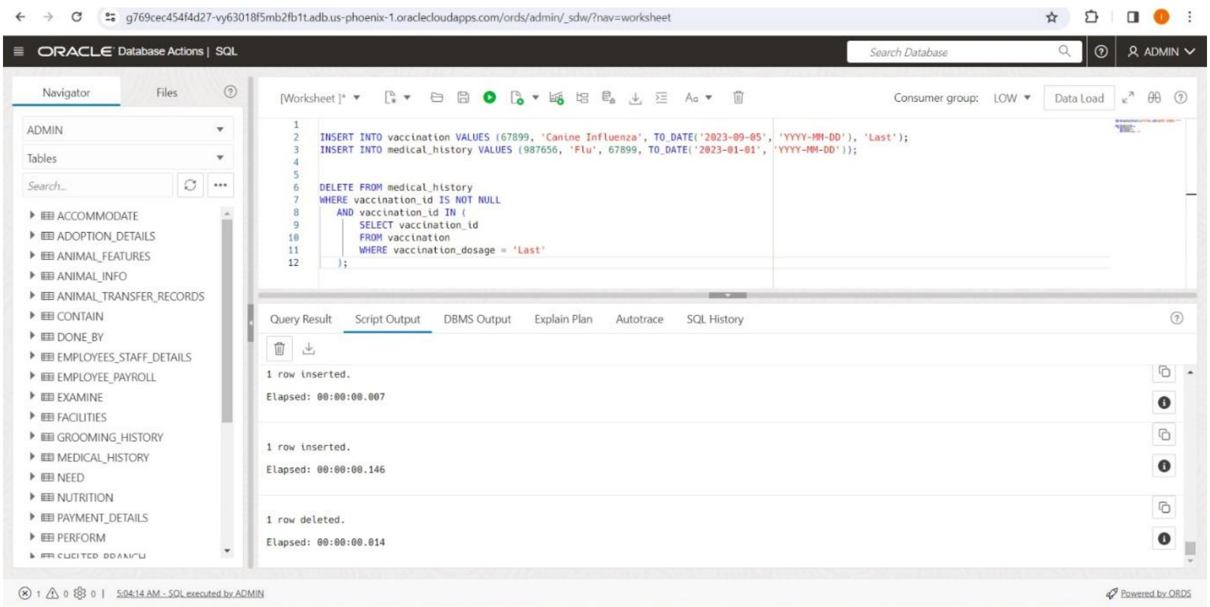

livesql.oracle.com/apex/f?p=590:1:28305192297254::RP::
Live SQL
Feedback
Help
lakshmikeerthigopireddy@my.unt.edu
SQL Worksheet
Clear
Find
Actions
Save
Run

```sql
INSERT INTO animal_info VALUES (9990, 'Cat', 'Labrador Retriever', DATE '2019-03-15', DATE '2023-05-01', 'Up to date', 1);
INSERT INTO animal_transfer_records VALUES (990, TO_DATE('2023-01-10', 'YYYY-MM-DD'), 'Shelter C', 'Shelter F', 9990);

DELETE FROM animal_transfer_records
WHERE animal_id IN (
    SELECT animal_id
    FROM animal_info
    WHERE a_type = 'Cat'
)
AND transfer_from = 'Shelter C';
```

```
1 row(s) inserted.

1 row(s) inserted.

1 row(s) deleted.
```

2024 Oracle · Live SQL 24.1.2, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 · Database Documentation · Ask Tom · Dev Gym
Built with ❤ using Oracle APEX · Privacy · Terms of Use

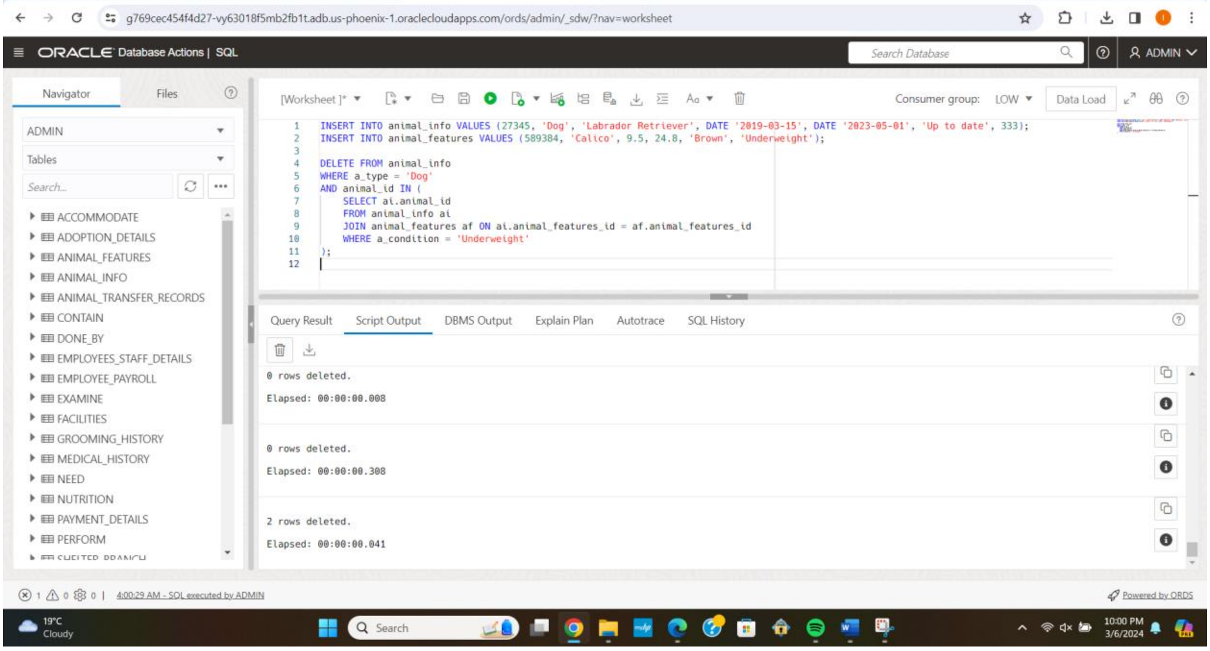
5. Delete medical history of the records where the vaccination dosage is last



```sql

INSERT INTO vaccination VALUES (67899, 'Canine Influenza', TO_DATE('2023-09-05', 'YYYY-MM-DD'), 'Last');
INSERT INTO medical_history VALUES (987656, 'Flu', 67899, TO_DATE('2023-01-01', 'YYYY-MM-DD'));


DELETE FROM medical_history
WHERE vaccination_id IS NOT NULL
  AND vaccination_id IN (
      SELECT vaccination_id
      FROM vaccination
      WHERE vaccination_dosage = 'Last'
  );
```

```
1 row inserted.

Elapsed: 00:00:00.007


1 row inserted.

Elapsed: 00:00:00.146


1 row deleted.

Elapsed: 00:00:00.014
```

6. Delete the animal information of Dogs which are underweight:



```sql
INSERT INTO animal_info VALUES (27345, 'Dog', 'Labrador Retriever', DATE '2019-03-15', DATE '2023-05-01', 'Up to date', 333);
INSERT INTO animal_features VALUES (589384, 'Calico', 9.5, 24.8, 'Brown', 'Underweight');

DELETE FROM animal_info
WHERE a_type = 'Dog'
AND animal_id IN (
    SELECT ai.animal_id
    FROM animal_info ai
    JOIN animal_features af ON ai.animal_features_id = af.animal_features_id
    WHERE a_condition = 'Underweight'
);
```

```
0 rows deleted.

Elapsed: 00:00:00.008


0 rows deleted.

Elapsed: 00:00:00.308


2 rows deleted.

Elapsed: 00:00:00.041
```

Individual Contributions:

My contribution revolved around a series of questions, each with a specific objective.

First, I addressed the issue of finding adoption-ready dogs in different locations. This was not just about the statistics. It was about integrating the creature into loving homes, ensuring that no dog was overlooked. After carefully entering and analysing detailed records, the query is written that allows us to determine which dogs are ready to start their new lives and where they are located.

A healthy diet comes next. Each animal has unique needs, and ensuring they are met with their necessities to their well-being. The query that developed to calculate total food requirements involves more than just math. It was about giving each animal the proper care and attention, and making sure they were healthy, happy and ready for adoption.

The next query is the environmental conditions that directly affect the lives of the animals. It was important to know which of our accommodations had play areas and which does not have temperature control. This effort was to ensure that our facilities were not just adequate but met the physical and emotional needs of the shelters.

Understanding the adoption schedule was also very important. The faster we can move animals from arrival to adoption, the more lives we can save. The query I did to find out the average length of time between acquiring and rescuing an animal. It helps us to know the period from arrival and adoption time.

Finally, examining organizational structure was about understanding the dynamics between managers and their teams to ensuring that the human resources were very well matched. This was to create a supportive, high-performing environment for staff and enabling them to do their best work for the animals. And each line of SQL isn’t just code, It was a stepping stone to a better future for countless animals.