

**PAWSOME, INC**   
Queries, Populating & Creating Pipeline to Cloud Pet Sitting Database

**BI Project ~**   
Date: May 22, 2024  
Authors: Pooja Patil

## **Section I: SQL Queries**

1. **Query 1 - Top 10 Highest Earning Sitters**
   1. **Description:** 
      1. This SQL query retrieves the sitters who have earned the most money by summing up their transaction amounts in the booking table. It can be used for payroll to determine the highest paid employees.
   2. **SQL:**

SELECT sitter.sitter\_id,

sitter.sitter\_firstname,

sitter.sitter\_lastname,

SUM(booking.transaction\_amount) as total\_earned

FROM sitter

INNER JOIN booking

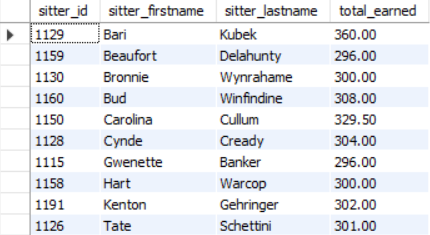
ON sitter.sitter\_id = booking.sitter\_id

GROUP BY sitter.sitter\_id

ORDER BY total\_earned desc

LIMIT 10;

* 1. **Output:**



1. **Query 2 - Owners with Unpaid Bills**
   1. **Description:**
      1. This SQL query pulls the owners who have unpaid bills. It lists the owners in order of who has the largest outstanding amounts due, so the billing department can determine who should be contacted to pay their outstanding bills.
   2. **SQL:**

SELECT o.owner\_id,

o.owner\_firstname,

o.owner\_lastname,

SUM(b.transaction\_amount) as amount\_due

FROM `owner` as o

INNER JOIN booking as b

ON o.owner\_id = b.owner\_id

INNER JOIN `schedule` as s

ON b.schedule\_id = s.schedule\_id

WHERE b.transaction\_status = 'UNPAID'

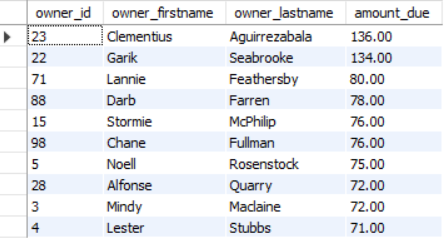
AND s.schedule\_end\_time < NOW()

GROUP BY owner\_id

ORDER BY amount\_due DESC

LIMIT 10;

* 1. **Output:**



1. **Query 3 - Pet Species by Hours Watched**
   1. **Description:**
      1. The query calculates the total amount of hours that each pet species is watched. Then, it calculates the percentage of hours each pet species is watched, so the company can examine what types of pets the business relies on.
   2. **SQL:**

SELECT DISTINCT

pet.pet\_species,

hours\_by\_species.hours\_watched,

(hours\_by\_species.hours\_watched / total\_watched.total\_hours) as perc\_of\_hours\_watched

FROM pet

INNER JOIN (

SELECT pet\_species,

SUM((time\_to\_sec(timediff(schedule\_end\_time, schedule\_start\_time))/3600)) as hours\_watched

FROM pet

INNER JOIN `schedule`

ON pet.pet\_id = `schedule`.pet\_id

INNER JOIN booking

ON `schedule`.schedule\_id = booking.schedule\_id

GROUP BY pet\_species

) hours\_by\_species

INNER JOIN (

SELECT SUM((time\_to\_sec(timediff(schedule\_end\_time, schedule\_start\_time))/3600)) as total\_hours

FROM pet

INNER JOIN `schedule`

ON pet.pet\_id = `schedule`.pet\_id

INNER JOIN booking

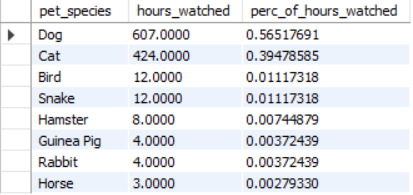
ON `schedule`.schedule\_id = booking.schedule\_id

) as total\_watched

WHERE hours\_by\_species.pet\_species = pet.pet\_species

ORDER BY perc\_of\_hours\_watched DESC;

* 1. **Output:**



1. **Query 4 - The Dogs Being Watched the Most**
   1. **Description:**
      1. The SQL query sums up the number of hours each pet is watched and returns the pets in order of highest number of hours. It allows the company to see the dogs that are continuing to use the service or their top clients.
   2. **SQL:**

SELECT pet.pet\_id,

pet.pet\_name,

SUM((time\_to\_sec(timediff(schedule\_end\_time, schedule\_start\_time))/3600)) as hours\_watched

FROM pet

INNER JOIN `schedule`

ON pet.pet\_id = `schedule`.pet\_id

INNER JOIN booking

ON `schedule`.schedule\_id = booking.schedule\_id

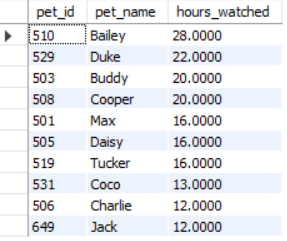
WHERE pet\_species like '%Dog%'

GROUP BY pet.pet\_id

ORDER BY hours\_watched DESC

LIMIT 10;

* 1. **Output:**



1. **Query 5 - Sitters with the Highest Ratings**
   1. **Description:**
      1. The query uses the average and count function to return the sitters who have the highest average rating, and it only includes sitters who have more than one rating. This query could be used by customers to find the best sitters for their pets or for the company to find the top performing employees.
   2. **SQL:**

SELECT sitter.sitter\_id,

sitter.sitter\_firstname,

sitter.sitter\_lastname,

AVG(review.review\_rating) as average\_rating,

COUNT(review.review\_id) as number\_of\_reviews

FROM sitter

INNER JOIN review

ON sitter.sitter\_id = review.sitter\_id

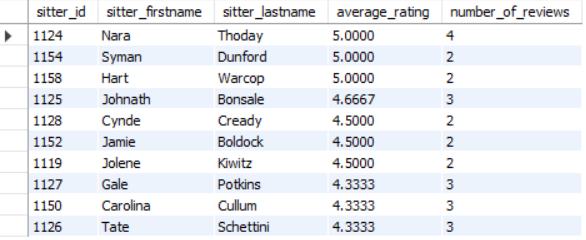
GROUP BY sitter.sitter\_id

HAVING COUNT(review.review\_id) > 1

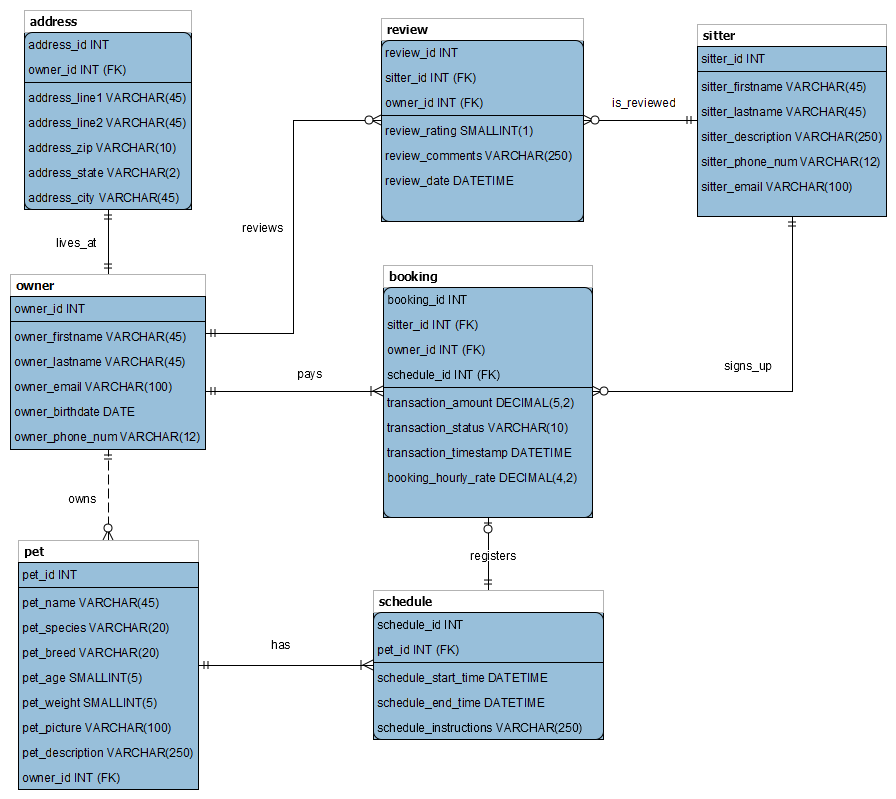
ORDER BY average\_rating DESC, number\_of\_reviews DESC

LIMIT 10;

* 1. **Output:**



## **Section II: ER Model**



## **Section III: Milestone 1 - Plan Implementation of Pet Sitting Database**

**Introduction:**

In today’s fast-paced world, pet ownership is on the rise, with more individuals and families welcoming pets into their homes but struggling to have quality time with their pets. Thus, there comes a growing need for reliable and efficient pet care services. Pet Sitters recognizes this market opportunity and aims to capitalize on it by building a pet sitting database to pair professional pet sitters with households requiring pet sitting services. This report outlines the rationale behind this project and highlights the benefits it will bring to the company.

**Rising Demand for Pet Sitting Services:**

The demand for pet sitting services is steadily increasing as pet owners seek trustworthy and professional care for their beloved companions. According to industry research, the pet care services market is experiencing significant growth because pet owners prefer hiring pet sitters instead of the more traditional pet care options available. Some reasons cited for using a pet sitter are to prevent stress to the animal caused by a changing environment, travel trauma, contracting illnesses and parasites from exposure to other animals.

According to the American Veterinary Medical Association, the number of dogs exceeded 80 million, and the number of cats exceeded 60 million in the U.S. in 2023. At the same time, the percentage of households owning dogs increased from 38% in 2016 to 45% in 2023. Households owning cats slightly increased from 25% in 2016 to 26% in 2023. By investing in a pet sitting database, Pet Sitters can position itself as a leader in this thriving industry and cater to the evolving needs of pet owners.

**Personalized Customer Experience:**

A pet sitting database will enable us to offer a more personalized and tailored experience to our customers. By maintaining comprehensive profiles of each pet and their respective owners, we can capture important details such as dietary preferences, and behavioral traits. This information will enable our pet sitters to provide customized care that meets the unique needs of every pet, fostering stronger relationships with both pets and their owners. Additionally, the database will enable us to track and analyze customer feedback, allowing us to continuously improve the quality of our services and revise our training programs of professional pet sitters.

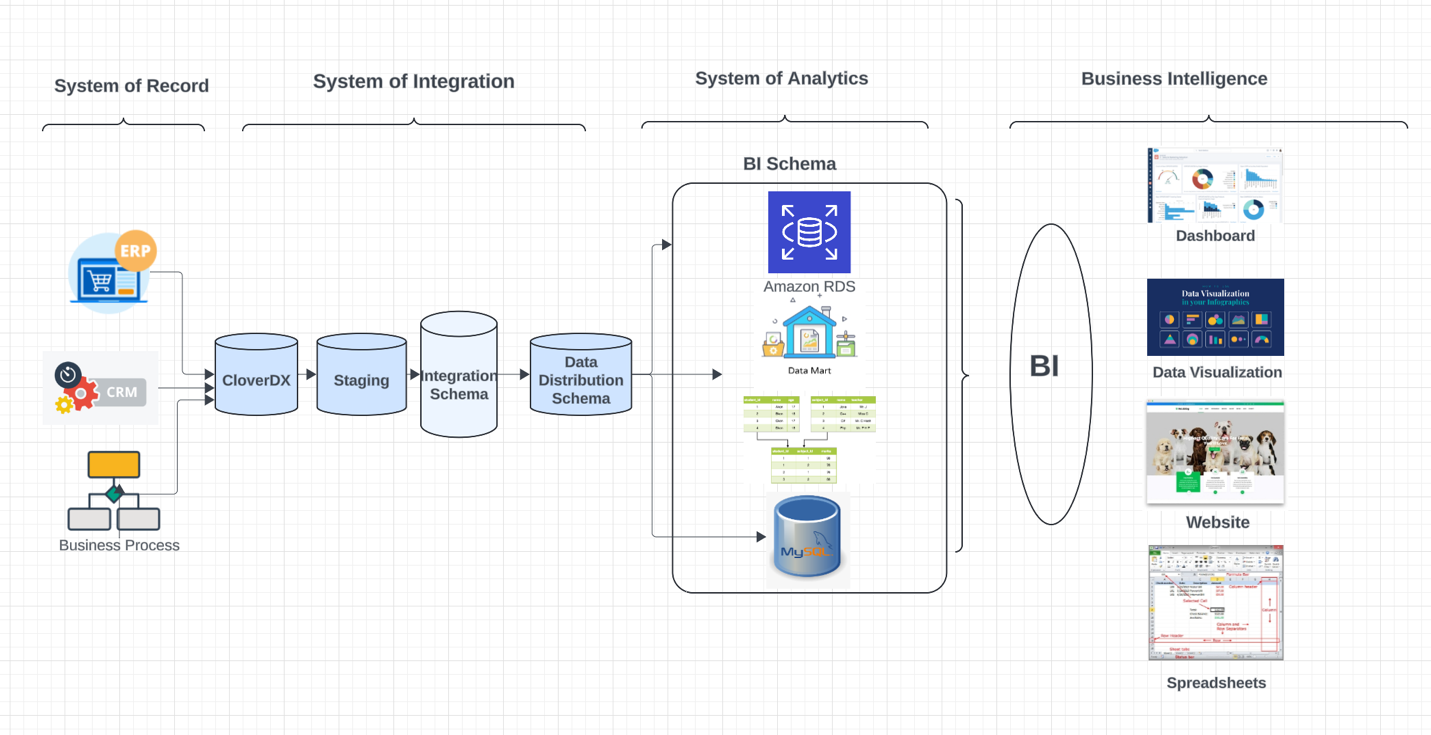
**Data-Driven Decision Making:**

The flow chart diagram below shows the design of our database architecture. From our discussions with Pet Sitters, we understand that the company has its on-premises enterprise resource planning (ERP) systems that help it manage its day-to-day business activities, such as accounting, finance, procurement, and human resources. The human resources systems help the company manage its employees, pet sitters and the review related to each pet sitter. In addition, the company is also using customer relationship management (CRM) systems to support its front-office business functions, such as marketing, sales, advertising, and customer service related to pet owners.

A diagram of a diagram of a data flow

Description automatically generated

**Updated:**



As the number of pets requiring pet sitting increases, there is a strong need to integrate their CRM, ERP, and other systems to effectively share data, rather than maintaining several separate sets of data. With a centralized data warehouse, it will be convenient for the company’s analysts to generate reports that can help the management team to make informed decisions. For example, we can find owner cities with the most open schedules for sitters to sign up. We can also show the best rated sitters each year on a dashboard. Moreover, historical data will be stored in our BI system, so trends over time can be presented in reports, such as the top earning sitters or the top watched pets.

**Conclusion:**

In conclusion, implementing a pet sitting database is a strategic investment that aligns with the growing demand for pet care services and the evolving needs of pet owners. By personalizing customer experiences, and leveraging data-driven insights, Pet Sitters can establish itself as a trusted provider of high-quality pet sitting services. This project is not just about creating a database; it's about laying the foundation for long-term success and growth in the dynamic pet care industry.

## **Section IV: RDS Connection Details**

We have uploaded three tables to our RDS database. The owner table was joined with the address table before loading it to our database. The other two tables are sitter and pet tables. Connection to our RDS database can be established using the following log-in information.

**Link:** database-1.cw2mnjnzhiu0.us-east-1.rds.amazonaws.com

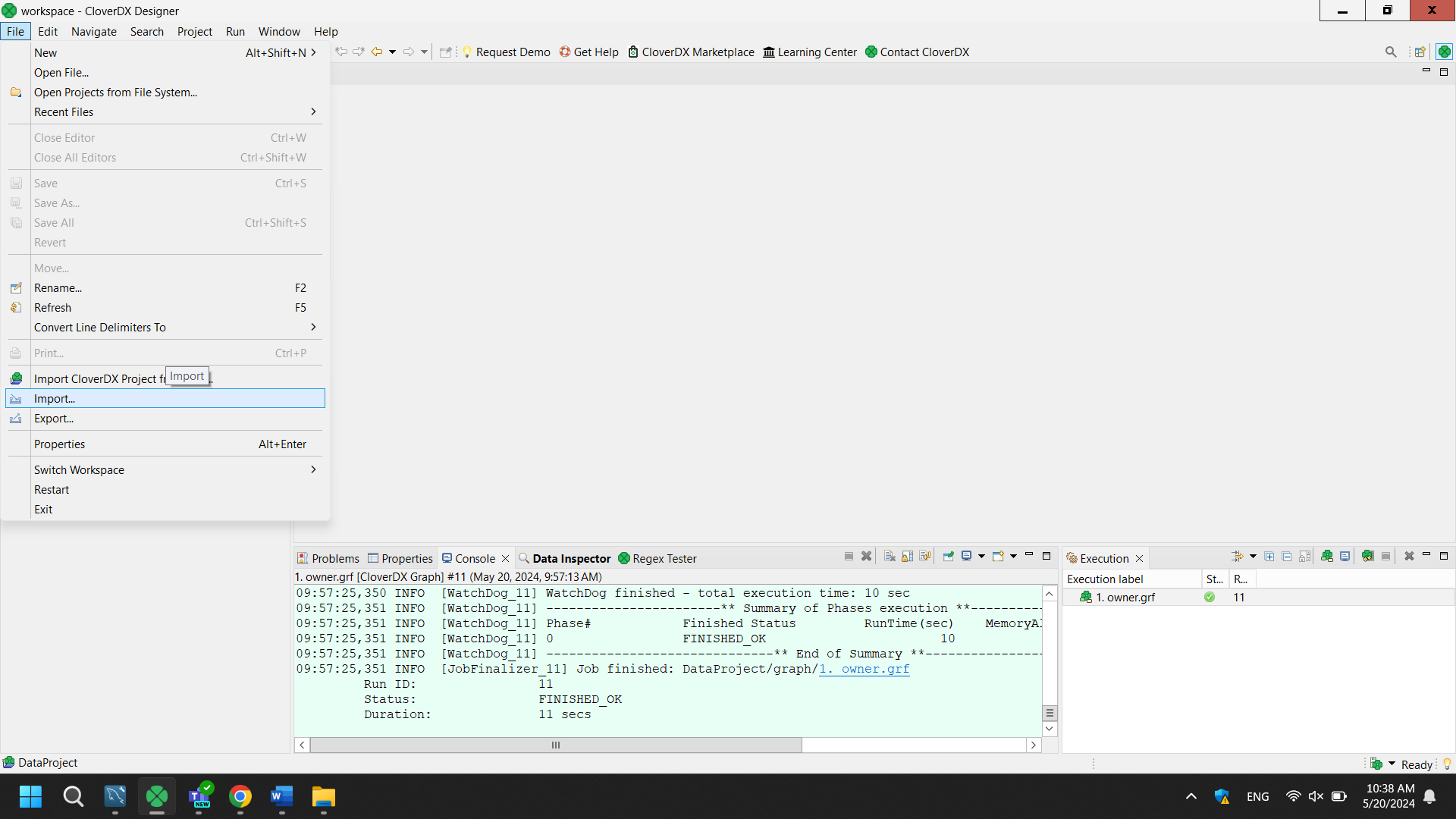
**Username:** admin

**Password:** adminadmin

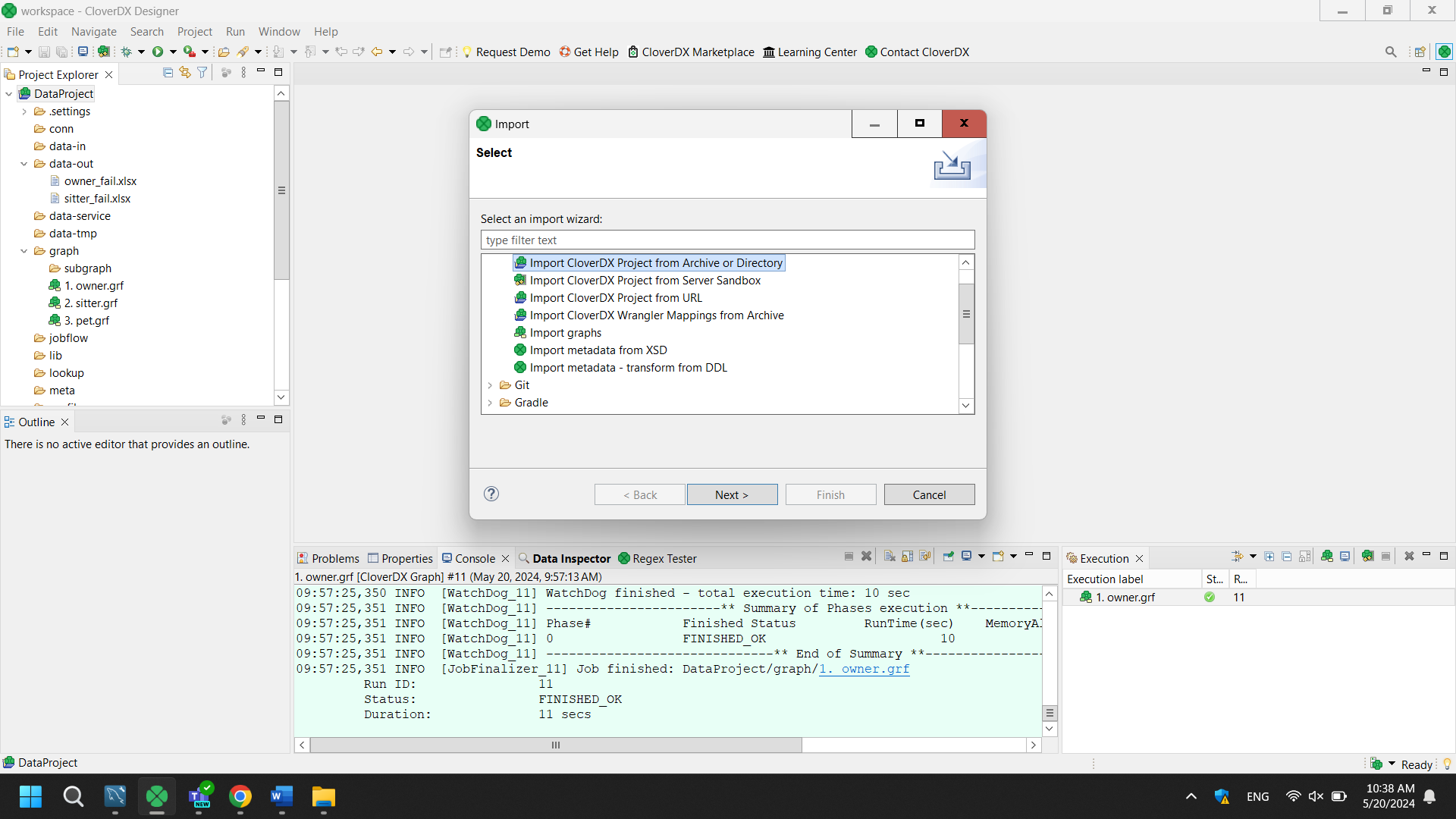
## **Section V: CloverDX**

To run CloverDX to move our data from cssql database to Amazon RDS database, please follow the following steps. Make sure you have downloaded our CloverDX project named ‘CloverDX\_Team4’ and save it on your computer.

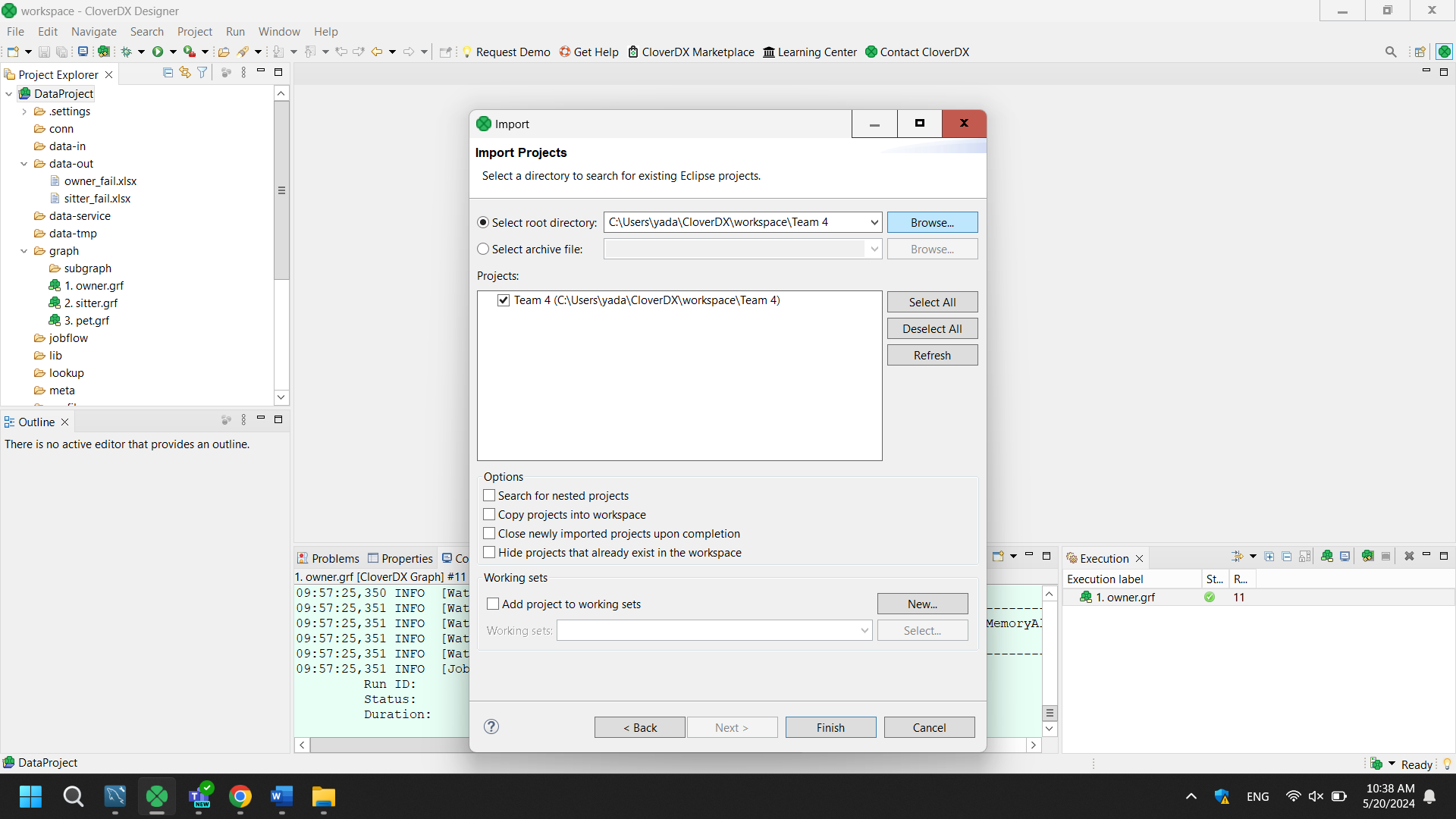
Step 1: Select File and then choose Import.



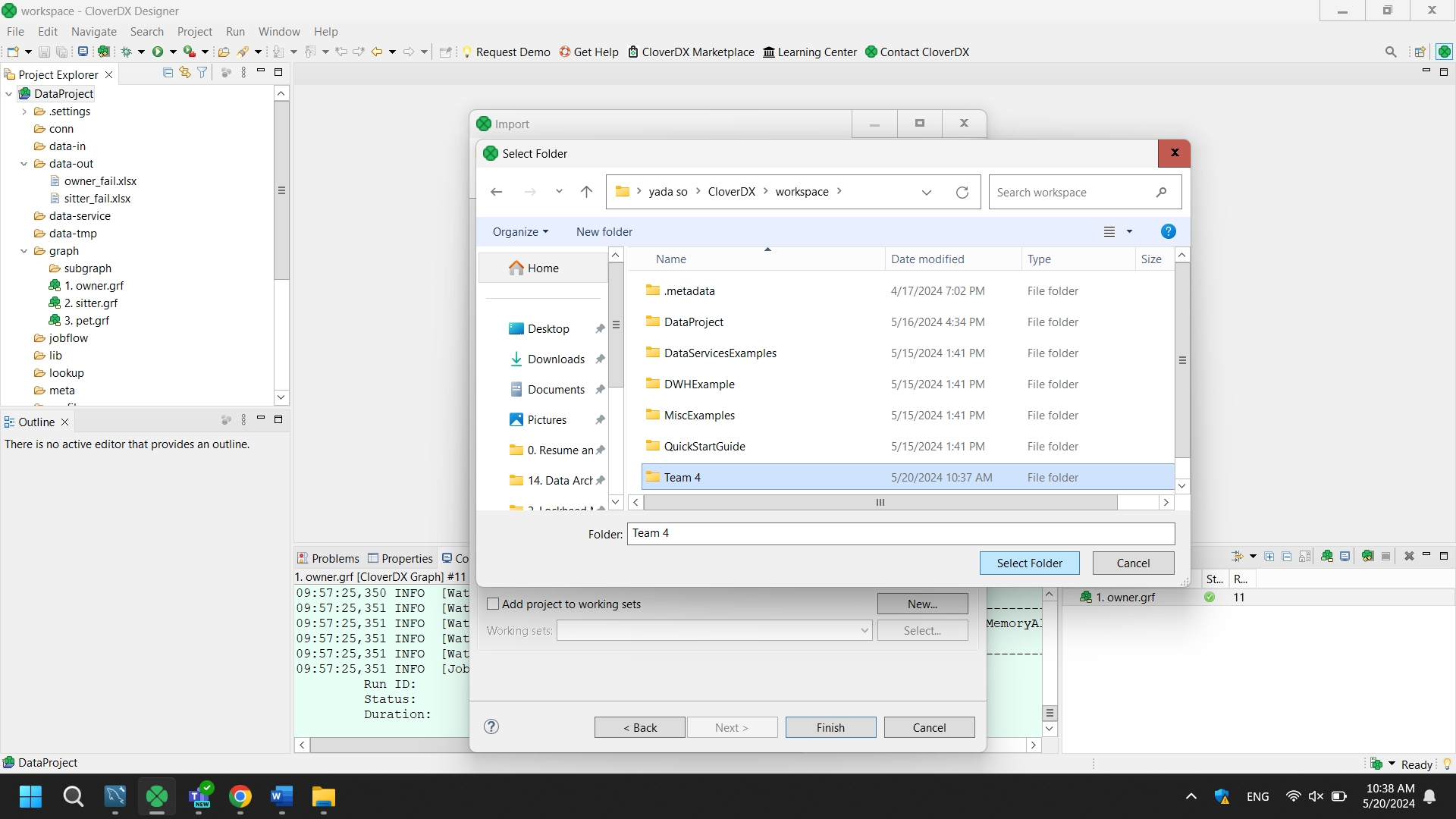
Step 2: On the Import window, select ‘Run CloverDX Project from Archive or Directory’ and then click ‘Next’.



Step 3: A new import window will pop up and then click ‘Browse’.



Step 4: After clicking the ‘Browse’ button, find the folder named ‘CloverDX\_Team4’ that keeps our CloverDX project on your computer, and then click finish.



Step 5: After importing ‘CloverDX\_Team4’ project to CloverDX, you should see 3 graphs in the ‘Project Explorer’ section for moving 3 tables in cssql server database to our RDS database. Select any graph you want, and you can run it to move the data. However, since we already loaded the data to Amazon RDS, you must delete the data from RDS first, or you will get an error.

A screenshot of a computer

Description automatically generated

1. **Owner Table**

A screenshot of a computer

Description automatically generated

The above graph shows our Clover DX pipeline for the owner table. It was joined with the address table using owner\_id before loading the data to our RDS database. We also checked if there were any duplicates and removed two duplicates from the owner table. Two owners younger than 18 years old were removed. Phone numbers in the existing owner table were not consistent, where we found that some phone numbers were written with hyphen such as 123-456-7890, some with space (123 456 7890), and some without space (1234567890). On CloverDX, we standardized our phone numbers using the format with hyphen (123-456-7890).

1. **Sitter Table**

A diagram of a computer

Description automatically generated

We removed 2 duplicates and reformat sitter’s phone numbers as explained above before loading the data to RDS database. To keep records of any removed data, we stored it in a separate spreadsheet.

1. **Pet Table**

A computer screen shot of a diagram

Description automatically generated

In our pet records, we initially added a transformer to eliminate duplicates. We checked for records with the same pets associated with the same owners. For our analysis, we stored the removed duplicates in a spreadsheet. Since our data had spaces in the string attributes, we implemented transformation logic to trim the leading and trailing spaces.