**SUPERVET PROJECT**

**MILESTONE 2**

KHANH LE

SAM FARMER

JOU HO

DUC VO

**SQL QUERIES**

**1. Display all prescriptions and medicine for all pets of a customer, looked up by customer's email.**

*-- Help retrieve medication history for all customer's pets*

SELECT prescriptionID,

Pet.petID,

Pet.name AS `pet\_name`,

Product.productID,

Product.name AS `drug\_name`,

PrescriptionProduct.quantity,

PrescriptionProduct.unit,

PrescriptionProduct.description,

Prescription.createdAt AS `prescribed\_date`

FROM Prescription

JOIN PrescriptionProduct USING (prescriptionID)

JOIN Product USING (productID)

JOIN Pet USING (petID)

JOIN Customer USING (customerID)

WHERE Customer.email = “dbevanslp@feedburner.com”

ORDER BY Pet.name;

Table

Description automatically generated

**2. Calculate Total Prescription Cost for each pet of a customer,**

**provided customerID**

*-- Help retrieve useful total cost data for customer’s pets*

SELECT petID, Pet.name AS `pet\_name`,

ROUND(SUM(PrescriptionProduct.quantity \* Product.price), 2)

AS `Total Prescription Cost`

FROM Prescription

JOIN PrescriptionProduct USING (prescriptionID)

JOIN Product USING (productID)

JOIN Pet USING (petID)

JOIN Customer USING (customerID)

WHERE customerID = 1781

GROUP BY petID

Background pattern

Description automatically generatedORDER BY Pet.name;

**3. Find out average prescription cost for a pet of customers who live in Washington state**

*-- This statistic helps with target prescription cost range for specific area*

SELECT ROUND(AVG(prescription\_cost), 2)

AS `Average Washington Prescription Cost` FROM (

SELECT SUM(Product.price \* PrescriptionProduct.quantity)

AS prescription\_cost

FROM Customer

JOIN Pet USING (customerID)

JOIN Prescription USING (petID)

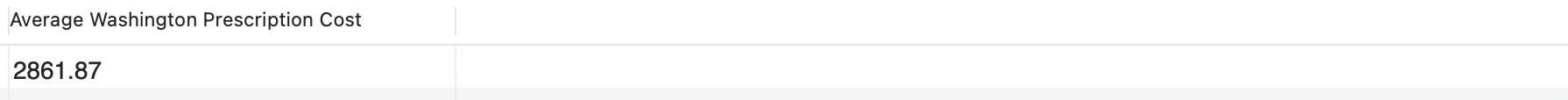
JOIN PrescriptionProduct USING (prescriptionID)

JOIN Product USING (productID)

WHERE state = “Washington”

GROUP BY prescriptionID

) AS PrescriptionCost;



**4. Find out customers who paid for the above average price product in store and when they bought it**

*-- This will help with marketing strategy such as send out ads or promotion.*

SELECT customerID, first\_name, last\_name, email,

Product.name as `product\_name`, Product.price,

SaleProduct.quantity,

SaleProduct.createdAt AS `purchased\_date`

FROM Customer

JOIN Payment USING (customerID)

JOIN SaleProduct USING (saleID)

JOIN Product USING (productID)

WHERE productID IN (

SELECT productID FROM Product

WHERE price > (SELECT AVG(price) FROM Product)

)

ORDER BY Product.price DESC;

Table

Description automatically generated with medium confidence

**5. This query will return a list of vets whose login password is less than 8 characters and does not contain special characters (alphanumerics)**

SELECT vetID, first\_name, last\_name, email

FROM Vet

WHERE LENGTH(password) < 8 AND password NOT LIKE '%[^a-zA-Z0-9]%';

A picture containing table

Description automatically generated

**6. This query will return a list of proucts and their total sale price in descending order**

SELECT productID, name, price, SP.quantity, category,

(price \* SP.quantity) AS totalsale

FROM SaleProduct SP

JOIN Product P USING (productID)

ORDER BY totalsale DESC;

*Table

Description automatically generatedOutput (showing first 10):*

​

**7. This query will return a list of states where the customers reside, and the total number of customers in each state in descending order**

SELECT state, COUNT(state) AS totalcustomer

FROM Customer

GROUP BY state

ORDER BY totalcustomer DESC;

*Background pattern

Description automatically generatedOutput (showing first 10):*

**8. this query will return the appointment ID, date, appointment status, vet ID, pet ID, and customer ID in December, 2008**

SELECT appointmentID, date, status, vetID, petID, customerID

FROM Appointment

WHERE YEAR(date) = 2008 AND MONTH(date) = 12

ORDER BY date;

Graphical user interface, table

Description automatically generated

**9. this query will return the total revenue, total pending amount, total paid amount, and total unpaid amount.**

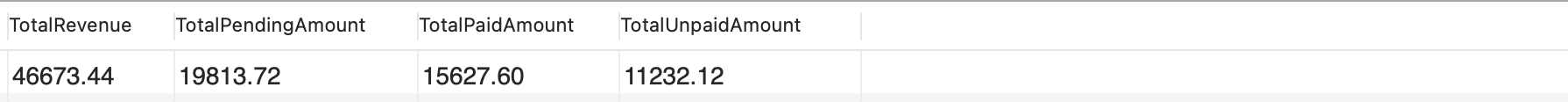
SELECT SUM(amount) AS TotalRevenue,

(SELECT SUM(amount) FROM Payment WHERE status = "pending") AS TotalPendingAmount,

(SELECT SUM(amount) FROM Payment WHERE status = "paid") AS TotalPaidAmount,

(SELECT SUM(amount) FROM Payment WHERE status = "unpaid") AS TotalUnpaidAmount

FROM Payment;



**10. this query will return the information of completed and prescribed appointments (customer ID, pet name, appointment date, doctor in charge), the total bill as well as the payment status of the prescription, with a provided customer ID.**

SELECT Customer.customerID AS CustomerID,

Pet.name AS PetName, Customer.email AS CustomerEmail,

Appointment.date AS AppointmentDate, Appointment.status AS AppointmentStatus,

Vet.last\_name AS Doctor, Prescription.status AS PrescriptionStatus,

ROUND((PrescriptionProduct.quantity \* Product.price), 2) AS TotalBill,

Payment.status AS PaymentStatus

FROM Customer

JOIN Appointment USING (customerID)

JOIN Pet USING (customerID)

JOIN Vet USING (vetID)

JOIN Prescription USING (vetID)

JOIN PrescriptionProduct USING (prescriptionID)

JOIN Product USING (productID)

JOIN Payment USING (prescriptionID)

WHERE Customer.customerID = 1005

AND Appointment.status = "completed"

AND Prescription.status = "prescribed";

Table

Description automatically generated with low confidence

**STORED PROCEDURE**

*-- This stored procedure retrieves a pet prescribed medication history with provided petID*

CREATE DEFINER=`mm\_cpsc502101team01`@`%`

PROCEDURE `pet\_medication\_history`(inputPetID int)

BEGIN

SELECT prescriptionID,

Pet.name AS `pet\_name`,

Product.productID,

Product.name AS `drug\_name`,

PrescriptionProduct.quantity,

PrescriptionProduct.unit,

PrescriptionProduct.description,

Prescription.createdAt AS `prescribed\_date`

FROM Prescription

JOIN PrescriptionProduct USING (prescriptionID)

JOIN Product USING (productID)

JOIN Pet USING (petID)

JOIN Customer USING (customerID)

WHERE Pet.petID = inputPetID;

END

**/\* Call the procedure for petID 1996 to retrieve prescibed medication history \*/**

CALL pet\_medication\_history(1996);

Table

Description automatically generated

**PHYSICAL MODEL ER DIAGRAM**

****