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

```sql
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;
```

prescriptionID	pet_na... ^	productID	drug_name	quantity	unit	description	prescribed_ date
1209	Felix	1112	Methylphenidate Hyd...	12	gram	Tadorna tadorna	2021-07-23 22:57:35
1209	Felix	1272	Naproxen Sodium	3	bottle	Cygnus buccinator	2021-07-23 22:57:35
1405	Felix	1002	CANDIDA ALBICANS	7	package	Alcelaphus buselaphus cokii	2020-12-18 18:48:44
1461	Felix	1037	Trazodone Hydrochlo...	8	mil	Phalacrocorax carbo	2021-04-05 19:30:04
1461	Felix	1119	Mucor plumbeus	14	count	Boa constrictor mexicana	2021-04-05 19:30:04
1336	Molly	1229	Clavamox	13	mg	Varanus sp.	2021-07-13 11:10:37
1336	Molly	1239	Drontal Feline	4	mg	Rana sp.	2021-07-13 11:10:37
1336	Molly	1285	Prednisone	13	bottle	Pycnonotus nigricans	2021-07-13 11:10:37
1336	Molly	1384	Onsior	4	bottle	Microcebus murinus	2021-07-13 11:10:37
1336	Molly	1482	Chaetomium globosum	18	mil	Castor canadensis	2021-07-13 11:10:37
1336	Molly	1599	Asoka	19	gram	Plocepasser mahali	2021-07-13 11:10:37
1161	Smudge	1265	Ranitidine	10	mg	Myrmecophaga tridactyla	2021-10-22 14:12:26
1161	Smudge	1448	Miconazole Nitrate	8	mil	Coendou prehensilis	2021-10-22 14:12:26
1161	Smudge	1512	Fixflex	16	package	Mabuya spilogaster	2021-10-22 14:12:26
1435	Sooty	1167	Spironolactone	10	mg	Sagittarius serpentarius	2020-11-30 15:22:21
1853	Sooty	1103	Ibuprofen	8	mil	Felis wiedi or Leopardus weidi	2021-05-08 15:00:15

2. Calculate Total Prescription Cost for each pet of a customer, provided customerID
-- Help retrieve useful total cost data for customer's pets

```sql
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
ORDER BY Pet.name;
```

petID	pet_name	Total Prescription Cost
1996	Felix	5241.27
1486	Molly	12513.1
1320	Smudge	3258.06
1991	Sooty	1894.18

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;
```

Average Washington Prescription Cost	
2861.87	

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;
```

customerID	first_name	last_name	email	product_name	price	quantity	purchased_date	
1689	Ariela	Teare	atearej5@plala.or.jp	Voltsillam	398.58	1	2021-11-10 02:01:10	
1731	Debee	Checchi	dchecchikb@amaz...	Bitchip	373.13	15	2021-05-09 15:18:22	
1228	Johnnie	Mowbray	jmowbray6c@virgin...	Bytecard	357.24	12	2021-04-19 07:00:43	
1613	Orlan	Garrettson	ogarrettsonh1@joo...	Regrant	343.23	11	2021-01-19 14:53:14	
1110	Cody	Labon	clabon32@arstechn...	Domainer	338.94	9	2020-12-21 07:25:38	
1613	Orlan	Garrettson	ogarrettsonh1@joo...	Subin	314.15	4	2021-03-03 08:25:45	
1339	Celesta	Edgecombe	cedgecombe9f@eh...	Daltfresh	290.32	12	2021-07-09 14:28:54	
1530	Engracia	Huggard	ehuggardeq@tripod...	Y-find	262.48	8	2021-08-21 18:52:18	
1339	Celesta	Edgecombe	cedgecombe9f@eh...	Quo Lux	260.38	3	2021-03-14 20:19:44	
1228	Johnnie	Mowbray	jmowbray6c@virgin...	Flexidy	214.46	8	2021-05-01 11:40:20	
1228	Johnnie	Mowbray	jmowbray6c@virgin...	Mat Lam Tam	184.10	9	2020-11-22 18:13:34	

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

```sql
SELECT vetID, first_name, last_name, email
FROM Vet
WHERE LENGTH(password) < 8 AND password NOT LIKE '%[^a-zA-Z0-9]%';
```

vetID	first_name	last_name	email
1002	Cymbre	Tarquini	ctarquini1@amazonaws.com
1003	Miguelita	Bonett	mbonett2@walmart.com
1004	Marji	Bancroft	mbancroft3@webmd.com
1010	Ilka	Povlsen	ipovlsen9@bluehost.com
NULL	NULL	NULL	NULL

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

```sql
SELECT productID, name, price, SP.quantity, category,
    (price * SP.quantity) AS totalsale
FROM SaleProduct SP
JOIN Product P USING (productID)
ORDER BY totalsale DESC;
```

Output (showing first 10):

productID	name	price	quantity	category	totalsale
1746	Bitchip	373.13	15	Apparel	5596.95
1890	Fixflex	365.35	15	Food	5480.25
1611	Rank	381.33	14	Food	5338.62
1722	Bitwolf	365.76	14	Odor Control	5120.64
1878	Veribet	364.13	14	Odor Control	5097.82
1862	Domainer	338.22	15	Treats and Chews	5073.30
1737	Konklux	378.37	13	Apparel	4918.81
1836	Viva	390.54	12	Apparel	4686.48
1852	Bytecard	357.24	13	Apparel	4644.12
1989	Subin	356.57	13	Apparel	4635.41
1560	Rank	376.32	12	Accessories	4515.84

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

```sql
SELECT state, COUNT(state) AS totalcustomer
FROM Customer
GROUP BY state
ORDER BY totalcustomer DESC;
```

Output (showing first 10):

state	totalcustomer
California	105
Texas	103
Florida	74
New York	60
Virginia	46
Ohio	33
Georgia	32
Pennsylvania	31
Colorado	30
Washington	28
Illinois	28

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

```sql
SELECT appointmentID, date, status, vetID, petID, customerID
FROM Appointment
WHERE YEAR(date) = 2008 AND MONTH(date) = 12
ORDER BY date;
```

appointmentID	date	status	vetID	petID	customerID	
9192	2008-12-02 22:42:16	checked in	1002	1051	1083	
1344	2008-12-03 06:11:52	checked in	1006	1041	1089	
6160	2008-12-11 04:25:41	checked in	1004	1082	1025	
5561	2008-12-18 06:56:52	completed	1003	1040	1096	
NULL	NULL	NULL	NULL	NULL	NULL	

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

```sql
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;
```

TotalRevenue	TotalPendingAmount	TotalPaidAmount	TotalUnpaidAmount	
46673.44	19813.72	15627.60	11232.12	

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.

```sql
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";
```

CustomerID	PetName	CustomerEmail	AppointmentDate	AppointmentStatus	Doctor	PrescriptionStatus	TotalBill	Payment
1005	Zola	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	4574.7	paid
1005	Zola	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	745.65	paid
1005	Janaya	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	4574.7	paid
1005	Janaya	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	745.65	paid
1005	Aurelie	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	4574.7	paid
1005	Aurelie	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	745.65	paid
1005	Zola	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	1466.93	paid
1005	Janaya	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	1466.93	paid
1005	Aurelie	frowthorne5@hostgator.com	2015-07-18 01:32:24	completed	Tarquini	prescribed	1466.93	paid

STORED PROCEDURE

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

```sql
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 */

```sql
CALL pet_medication_history(1996);
```

prescriptionID	pet_name	productID	drug_name	quantity	unit	description	prescribed_ date
1209	Felix	1112	Methylphenidate Hyd...	12	gram	Tadorna tadorna	2021-07-23 22:57:35
1209	Felix	1272	Naproxen Sodium	3	bottle	Cygnus buccinator	2021-07-23 22:57:35
1405	Felix	1002	CANDIDA ALBICANS	7	package	Alcelaphus buselaphus cokii	2020-12-18 18:48:44
1461	Felix	1037	Trazodone Hydrochlo...	8	mil	Phalacrocorax carbo	2021-04-05 19:30:04
1461	Felix	1119	Mucor plumbeus	14	count	Boa constrictor mexicana	2021-04-05 19:30:04

PHYSICAL MODEL ER DIAGRAM

Customer
- 🔑 customerID INT
- ◇ createdAt TIMESTAMP
- ◇ address VARCHAR(45)
- ◇ address2 VARCHAR(45)
- ◇ city VARCHAR(45)
- ◇ state VARCHAR(45)
- ◇ zip_code VARCHAR(45)
- ◇ password VARCHAR(45)
- ◇ email VARCHAR(45)
- ◇ first_name VARCHAR(45)
- ◇ last_name VARCHAR(45)

Indexes

Payment
- 🔴 customerID INT
- 🔴 saleID INT
- 🔴 prescriptionID INT
- ◇ createdAt TIMESTAMP
- ◇ amount DECIMAL(10,2)
- ◇ status VARCHAR(45)
- ◇ method VARCHAR(45)

Indexes

Sale
- 🔑 saleID INT
- ◇ description VARCHAR(500)
- ◇ createdAt TIMESTAMP
- ◇ updatedAt TIMESTAMP
- ◆ employeeID INT

Indexes

SaleProduct
- 🔴 saleID INT
- 🔴 productID INT
- ◇ quantity INT
- ◇ description VARCHAR(45)
- ◇ createdAt TIMESTAMP

Indexes

Pet
- 🔑 petID INT
- ◇ createdAt VARCHAR(45)
- 🔴 customerID INT
- ◇ name VARCHAR(45)
- ◇ breed VARCHAR(45)

Indexes

Employee
- 🔑 employeeID INT
- ◇ createdAt TIMESTAMP
- ◇ first_name VARCHAR(45)
- ◇ last_name VARCHAR(45)
- ◇ email VARCHAR(45)
- ◇ address VARCHAR(45)
- ◇ address 2 VARCHAR(45)
- ◇ city VARCHAR(45)
- ◇ state VARCHAR(45)
- ◇ zip_code VARCHAR(45)

Indexes

Product
- 🔑 productID INT
- ◇ name VARCHAR(300)
- ◇ brand VARCHAR(300)
- ◇ price DECIMAL(10,2)
- ◇ color VARCHAR(45)
- ◇ size VARCHAR(5)
- ◇ quantity INT
- ◇ createdAt TIMESTAMP
- ◇ updatedAt TIMESTAMP
- ◇ description VARCHAR(500)
- ◇ category VARCHAR(45)

Indexes

Vet
- 🔑 vetID INT
- ◇ createdAt VARCHAR(45)
- ◇ password VARCHAR(45)
- ◇ email VARCHAR(45)
- ◇ first_name VARCHAR(45)
- ◇ last_name VARCHAR(45)

Indexes

Appointment
- 🔑 appointmentID INT
- ◇ date DATETIME
- ◇ status VARCHAR(45)
- 🔴 vetID INT
- 🔴 petID INT
- ◇ createdAt TIMESTAMP
- 🔴 customerID INT

Indexes

Prescription
- 🔑 prescriptionID INT
- 🔵 petID INT
- ◇ status VARCHAR(45)
- ◇ createdAt TIMESTAMP
- 🔴 vetID INT

Indexes

PrescriptionProduct
- 🔴 prescriptionID INT
- 🔴 productID INT
- ◇ quantity VARCHAR(45)
- ◇ unit VARCHAR(45)
- ◇ description VARCHAR(500)
- ◇ createdAt TIMESTAMP

Indexes