Lab 4: Aggregate Functions (6% of total grade)

Submission: Use the included .sql file to put your answers in, then upload only the PDF file to Blackboard (Assessments > Lab 4 - Aggregate Functions).

Name your file: HTTP5126-L4-AggFns-*LastNameFirstName*.pdf, replace *LastNameFirstName* with your name as displayed in Blackboard.

Purpose: To implement new keywords and clauses learned in Lesson 4 in order to provide grouped or, "aggregate", result sets.

Requirements: For this assignment, you will use the provided Pet Store data tables.

NOTE: Run your queries on your database to make sure desired results are retrieved. Also import and execute your sql file to ensure it runs all your queries before submitting.

Pre-Lab:

1. Start your mySQL server and open phpMyAdmin.
2. Feel free to reuse the database from lab 2 or create a new one for lab 4 using the **pet_store_tables.sql** file (same file as lab 2).

Part 1: MIN(), MAX(), AVG(), & SUM() (2%)

For each of the following, create a SQL query that returns the number requested.

A. "What is the lowest price of any item?"

```
SELECT MIN(price) FROM stock_item;
```

B. "What is the greatest quantity of any item in stock?"

```
SELECT MAX(inventory) FROM stock_item;
```

C. "What is the average price of all the items in the store?"

```
SELECT AVG(price) FROM stock_item;
```

D. "What is the total inventory of all the items in the store?"

```
SELECT SUM(inventory) FROM stock_item;
```

Part 2: Count 'Em By Groups (1.5%)

A. Provide a count of employees grouped by role. Include the role and the count in your results.

```
SELECT role, COUNT(employee_id) FROM employee GROUP BY role;
```

B. Create a count of items by category that excludes the fish category ("piscine"). Put 'Mammals' as the heading for the category.

```
SELECT category, COUNT(item) AS items FROM stock_item WHERE category
NOT IN ('piscine') GROUP by category;
```

C. Find the 'Average Price ($)' of items for each category. Include the average price and category in your results. Don't include items that are out of stock in your averages.

```
SELECT category, AVG(price) FROM stock_item WHERE inventory > 0 GROUP BY category;
```

Part 3: Groups & Having (1.5%)

A. Manager: "Inventory time. I need the total number of remaining stock for each animal category. Sort it by the fewest items to the most items. At the top of the chart I want to see 'In Stock', and 'Species'."

```
SELECT category AS Species, inventory As InStock FROM stock_item ORDER BY inventory ASC;
```

B. Manager: "Get the total stock and average price for each animal category. Include categories with fewer than 100 items in stock and an average price below 100."

```
SELECT category, AVG(price), SUM(inventory) FROM stock_item GROUP BY category HAVING COUNT(inventory) < 100 AND AVG(price) < 100;
```

Part 4: Challenging Calculations (1%)

A. Manager: "Calculate our 'Potential Earnings' for the remaining stock if we were to sell everything. Each of the 20 items should have its own row. Show me the 'Potential Earnings', 'Product', 'Price', 'Stock Remaining', and 'Species'. Display prices with a dollar sign (e.g., $55). Sort the table from highest to lowest 'Potential Earnings'."

```
SELECT (price * inventory) AS potential_earnings, item, CONCAT("$", price) AS price, inventory AS stock_remaining, category AS species FROM stock_item ORDER BY potential_earnings DESC;
```