ISOM 352 Midterm

Name: Christine Zhou
Email: christine.zhou@emory.edu

Question 1: (6 points, maximum 2-pages double-spaced)

1. (6 points) Answer the following short questions

1.1. Briefly describe M:N relationship between two entities.
This is a many to many relationship, meaning that entities on both sides of the relationship can have many related entities on the other side. Some examples are friends to family or restaurant food chains to locations.

1.2. Briefly describe when "WHERE" clause can and cannot be used to select rows after a "GROUP BY" clause?
Where cannot be used after a group by clause since the GROUP BY comes first in syntax. You must use "HAVING" instead.

1.3. Briefly describe the difference between INNER JOIN and CROSS JOIN?
INNER JOIN means only those rows that satisfy the join are included in the result, while CROSS JOIN joins each row from the first table with each row from the second table. The result set returned by a cross join is a Cartesian product.

1.4. Write a SQL select statement to convert a text date to datetime format using cast function (use text date: "2022-10-13")

```
SELECT CAST('2022-10-13' AS DATETIME);
```

1.5. Write a SQL select statement using regular expression (regexp) to show only those rows that have email addresses with only alphabets [a-z] in the email address and ending in ".com"
Assume that there exists a table 'customers' which has an email address column.

```
SELECT email_address FROM customers WHERE email_address REGEXP '^[A-Z@]+.com$';
```

1.6. Write a SQL select statement to find two NBA teams that have the largest difference in their scores after their first game. Assume the "score" table has columns team_name and score_game_1 (hint: you can join "score" table with itself)
Assign aliases to code

```
SELECT s1.team_name, s2.team_name, MAX(s1.score_game_1 - s2.score_game_1) AS difference
FROM score AS s1
CROSS JOIN score AS s2
GROUP BY s1.team_name
```

```
ORDER BY difference DESC
LIMIT 1;
```

Question 2: (6 points, one SQL code file, and maximum 1-page double-spaced)

2. Assuming you are the owner of the business (my_guitar_shop) and are interested in sending a loyalty promotion email to customers. Answer the following questions using my_guitar_shop database provided on canvas:

2.1. [2 points] Write an SQL statement that uses my_guitar_shop database (provided for mySQL) and selects customer name, email, number of products purchased, total purchase amount, and average discount_amount. [hint: you will need customer table, orders table, and order_items Table]

```
SELECT c.first_name, c.email_address, SUM(oi.quantity),
SUM((oi.item_price - oi.discount_amount) * oi.quantity) AS total_purchase,
AVG(oi.discount_amount) #select various insights
FROM customers AS c #from first table
INNER JOIN orders o ON c.customer_id = o.customer_id #join by columns
INNER JOIN order_items oi ON oi.order_id = o.order_id
GROUP BY c.customer_id; #aggregate by customer id
```

2.2. [2 points] Write an SQL statement that selects product_id, category_id, product_name, product_brand, total revenue, and total discount amount.

```
SELECT p.product_id, p.category_id, p.product_name, SUBSTRING_INDEX(p.product_name,'
', 1) AS product_brand,
SUM(p.list_price * (1 - (p.discount_percent / 100)) * oi.quantity) AS total_rev,
SUM(oi.quantity*oi.discount_amount) AS total_discount_amount #select various insights
FROM products AS p
INNER JOIN order_items oi ON oi.product_id = p.product_id #join by columns
GROUP BY p.product_id; #aggregate by product id
```

2.3. [2 point] Briefly discuss (maximum 1 page, double spaced) what product will you promote to what customer and offer what promotion ($off or %off) and why.

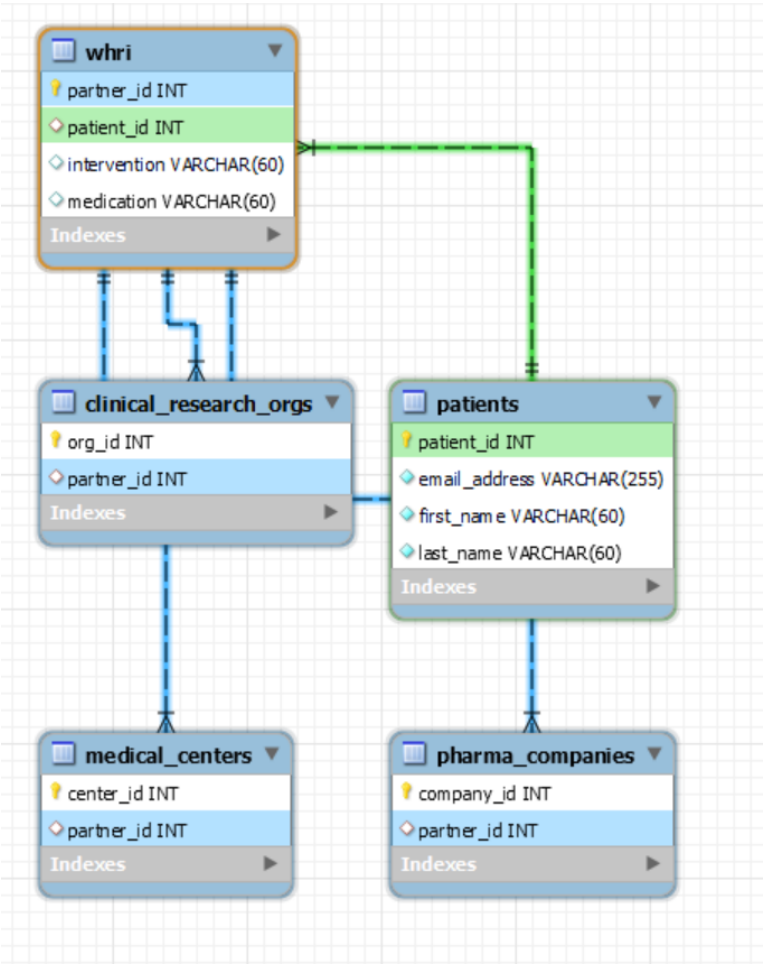
To decide what I would promote to customers, and which promotions I would use, I would first look at the analysis I have already conducted. I would analyse what products have the lowest total revenue, and consider a discount on the price of this product. The discount may boost sales for this product.

To target customers, I would look at which customers are spending the highest total purchase amount, then give them small discounts as part of a membership program so that they may spend even more. The purchase amount indicates their loyalty to the store.

Question 3: (6 points, maximum 2-pages double-spaced)

3. [6 points] Designing a Database: Recently Walmart announced the creation of Walmart Healthcare Research Institute (WHRI) to provide equitable access to high quality, affordable and convenient healthcare resources, including innovative research Walmart is working with a wide range of partners, including clinical research organizations, pharmaceutical companies, and medical centers.

3.1. [4 points] Create a database schema (entity-relationship model) for WHRI considering all data the company may need to provide equitable access to affordable access to patients. Please create tables in mySQL workbench with relationships and reverse engineer the ER model. Submit only the EER model. (Consider 1-3 tables each for patients, WHRI, clinical research organizations, pharmaceutical companies, and medical centers)



3.2. [2 points] Write a SQL query that would help WHRI select all information on a patient from the above schema.

```
USE whri;
SELECT * FROM patients AS p
INNER JOIN WHRI as w on p.patient_id = w.patient_id;
```

Question 4: (6 points, one Py code file only)

Please provide code examples (as short and simple as you can) that generates the following errors when executed (submit only one py file):

4.1. SyntaxError

```
print(My name is Christine)
```

4.2. NameError

```
hello()
```

4.3. TypeError

```
"hello" + 3
```

4.4. IndexError

```
letters = ["A", "B"]
letters[3]
```

4.5. IndentationError

```
if 1 == 1:
     print("This statement is true")
    print("This gives an indentation error")
```

4.6. ValueError

```
import math
math.sqrt(-1)
```

Question 5: (6 points, one Py code file only)

5. Sentiment analysis. Develop a sentiment analysis tool for Starbucks stores. Pick any Starbucks location on Google Maps and access their reviews.

5.1. [5 points] Steps for sentiment analysis:

5.1.1. Load the positive and negative words data using csv.reader into python. (Feel free to manually remove the header of the file)

```
# 5.1.1
import csv

pos_words = open("positive-words.csv", "r")
d = csv.reader(pos_words)
positive_w = list(d)

neg_words = open("negative-words.csv", "r")
c = csv.reader(neg_words)
negative_w = list(c)
```

5.1.2. Write two functions (numPositiveWords, numNegativeWords) that take a text input and returns the total number of positive and negative words in that text. (You will need to go over the entire list of words in each file you loaded in the previous step)

```
# 5.1.2
def numPositiveWords(text):
    split_text = text.split()
    count = 0
    for word in split_text:
        for i in positive_w:
            if word in i:
                count = count + 1
    return count

numPositiveWords(input("Please input text: "))


def numNegativeWords(text):
    split_text = text.split()
    count = 0
    for word in split_text:
        for i in negative_w:
            if word in i:
                count = count + 1
    return count

numNegativeWords(input("Please input text: "))
```

5.1.3. Write a function that takes a number of positive words, number of negative words, and total number of words in a string to return a sentiment score. Sentiment is calculated as: (“number of positive words” - “number of negative words”)/(total words).

```
#5.1.3
def sentimentScore(text):
#     text = input("Please input text: ")
    num_pos_words = numPositiveWords(text)
    num_neg_words = numNegativeWords(text)
    def word_count(string):
        return(len(string.strip().split(" ")))
```

```
    total_words = word_count(text)
    score = (num_pos_words - num_neg_words)/total_words
    return score

sentimentScore(input("Please input text: "))
```

5.1.4. Write a function that gets a review string from a user and validates that the string is less than 280 characters. If the string is longer than 280 characters, print a warning message and truncate the review to 280 characters.

```
#5.1.4
def check_length(string):
    if len(string) <= 280:
        print("The review is less than 280 characters in length.")
    else:
        print("The review is more than 280 characters in length, and will be truncated.")
        new_string = string[:280]
    return new_string

check_length(input("Please input text: "))
```

5.1.5. Print the number of positive words, negative words, and sentiment of the text.

```
#5.1.5
print(numPositiveWords(input("Please input text: ")))
print(numNegativeWords(input("Please input text: ")))
print(sentimentScore(input("Please input text: ")))
```

5.1.6. If the sentiment is positive, print some happy message. If sentiment is negative, print a sad message.

```
#5.1.6
def overall_sentiment(text):
    try:
        if float(sentimentScore(text)) > 0:
            print("Congrats, your score is above zero! Today the sun is shining brightly :)")
        elif float(sentimentScore(text)) < 0:
            print("Your score is below zero. The days ahead are dark indeed...")
        else:
            print("The review is not positive or negative.")
```

```
    except:
        Print("Please enter a valid string input.")

overall_sentiment(input("Please input text: "))
```

5.2. [1 point] briefly discuss (in the code) potential application of this tool, based on different reviews you tested, for Starbucks company.
The Starbucks chain could decide whether they have an overall positive or negative impression on their customers. They could run analyses on how many positive words, negative words, and overall sentiment scores they receive, and aggregate these numbers to see whether they are doing a great job on customer service. If they are getting a bad customer service score, they should consider better training for their employees.