Andrea Gray
CS 317
Homework #2

Nested Queries

Tables (customers, receipts, and students) were created within my database (graya17_db) for this assignment based off the .sql files given.

1. The first questions refer to the customers and receipts tables.
   a. Who were the 4 best customers based on total purchase dollar amount?

```
mysql> SELECT receipts.customer_id, first_name, last_name, SUM(receipts.amount) totalSpent
    -> FROM customers
    -> INNER JOIN receipts ON customers.customer_id = receipts.customer_id
    -> GROUP BY customer_id
    -> ORDER BY SUM(amount) DESC
    -> LIMIT 4;
+-------------+------------+-----------+------------+
| customer_id | first_name | last_name | totalSpent |
+-------------+------------+-----------+------------+
|           2 | Carl       | Orf       |    1713.26 |
|          10 | John       | Adams     |    1580.12 |
|           7 | Johann     | Bach      |    1500.92 |
|           5 | Richard    | Wagner    |    1434.76 |
+-------------+------------+-----------+------------+
4 rows in set (0.00 sec)
```

   b. Who were the 5 biggest spenders based on average purchase?

```
mysql> SELECT receipts.customer_id, first_name, last_name, AVG(receipts.amount) avgSpent
    -> FROM customers
    -> INNER JOIN receipts on customers.customer_id = receipts.customer_id
    -> GROUP BY customer_id
    -> ORDER BY AVG(amount) DESC
    -> LIMIT 5;
+-------------+------------+---------------+-----------+
| customer_id | first_name | last_name     | avgSpent  |
+-------------+------------+---------------+-----------+
|           6 | Ludwig     | van Beethoven | 46.639600 |
|          11 | Kurt       | Weill         | 44.922857 |
|           1 | Frank      | Ferguson      | 43.714194 |
|           3 | Leonard    | Bernstein     | 43.184828 |
|           9 | Vaghan     | Williams      | 42.291724 |
+-------------+------------+---------------+-----------+
5 rows in set (0.00 sec)
```

   c. Who were the 3 most frequent shoppers based on number of different purchases?

```
mysql> SELECT receipts.customer_id, customers.first_name, customers.last_name, COUNT(*) purchases
    -> FROM receipts
    -> INNER JOIN customers ON receipts.customer_id = customers.customer_id
    -> GROUP BY customer_id
    -> ORDER BY COUNT(*) DESC
    -> LIMIT 3;
+-------------+------------+-----------+-----------+
| customer_id | first_name | last_name | purchases |
+-------------+------------+-----------+-----------+
|           2 | Carl       | Orf       |        49 |
|          10 | John       | Adams     |        43 |
|           7 | Johann     | Bach      |        43 |
+-------------+------------+-----------+-----------+
3 rows in set (0.01 sec)
```

d. What was the busiest day?

```
mysql> SELECT purchase_date, COUNT(*)
    -> FROM receipts
    -> GROUP BY purchase_date
    -> ORDER BY COUNT(*) DESC
    -> LIMIT 1;
+---------------+----------+
| purchase_date | COUNT(*) |
+---------------+----------+
| 2018-05-27    |        6 |
+---------------+----------+
1 row in set (0.00 sec)
```

e. What was the most profitable day?

```
mysql> SELECT purchase_date, SUM(amount) profit
    -> FROM receipts
    -> GROUP BY purchase_date
    -> ORDER BY SUM(amount) DESC
    -> LIMIT 1;
+---------------+--------+
| purchase_date | profit |
+---------------+--------+
| 2018-07-27    | 241.70 |
+---------------+--------+
1 row in set (0.00 sec)
```

f. Who made purchases on the busiest day? (there were 6 purchases on that day)

```
mysql> SELECT receipts.customer_id, first_name, last_name
    -> FROM customers
    -> INNER JOIN receipts ON customers.customer_id = receipts.customer_id
    -> WHERE purchase_date = '2018-05-27';
+-------------+------------+-----------+
| customer_id | first_name | last_name |
+-------------+------------+-----------+
|          15 | Alban      | Berg      |
|           3 | Leonard    | Bernstein |
|          13 | John       | Cage      |
|          10 | John       | Adams     |
|          10 | John       | Adams     |
|           1 | Frank      | Ferguson  |
+-------------+------------+-----------+
6 rows in set (0.00 sec)
```

2. These questions refer to the students table.

a. How many sibling pairs are there? (Just having the same last name is not a sibling. They have the same home.)

```
mysql> SELECT street_address, last_name, city, COUNT(*)
    -> FROM students
    -> GROUP BY street_address
    -> HAVING COUNT(*) = 2;
+----------------+-----------+---------------+----------+
| street_address | last_name | city          | COUNT(*) |
+----------------+-----------+---------------+----------+
| 1940 Grant St. | Jones     | Denver        |        2 |
| 8046 Maple St. | Cook      | Highland Park |        2 |
+----------------+-----------+---------------+----------+
2 rows in set (0.00 sec)
```

b. Name all siblings.

```
mysql> SELECT first_name, last_name
    -> FROM students
    -> INNER JOIN (SELECT street_address
    -> FROM students
    -> GROUP BY street_address
    -> HAVING COUNT(*) = 2) AS siblings
    -> ON students.street_address = siblings.street_address;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Leonard    | Cook      |
| Sam        | Cook      |
| Mary       | Jones     |
| Phyllis    | Jones     |
+------------+-----------+
4 rows in set (0.00 sec)
```

c. How many twins are there? (Just having the same birthday does not make you twins.)

```
mysql> SELECT last_name, street_address, birthday
    -> FROM students
    -> GROUP BY birthday, street_address
    -> HAVING COUNT(*) = 2;
+-----------+-----------------+------------+
| last_name | street_address  | birthday   |
+-----------+-----------------+------------+
| Carlson   | 250 Pines Blvd. | 1999-07-06 |
| Jones     | 1940 Grant St.  | 1999-11-13 |
+-----------+-----------------+------------+
2 rows in set (0.00 sec)
```

d. Name the twins.

```
mysql> SELECT first_name, last_name
    -> FROM students
    -> INNER JOIN (SELECT birthday, street_address
    -> FROM students
    -> GROUP BY birthday, street_address
    -> HAVING COUNT(*) = 2) AS twins
    -> ON students.birthday = twins.birthday
    -> AND students.street_address = twins.street_address;
+------------+-----------+
| first_name | last_name |
+------------+-----------+
| Bill       | Carlson   |
| Jean       | Carlson   |
| Mary       | Jones     |
| Phyllis    | Jones     |
+------------+-----------+
4 rows in set (0.00 sec)
```