Assignment 9 - Subqueries (55 points)
Due Date: Wednesday, April 6, 2016 11:55 PM

Objectives:

The goal of the assignment is to get you to think about subqueries. A subquery is necessary when a search is based on an unknown value that is contained within a database table.

This assignment uses the tables associated with the *bookstore* database.

Write SQL statements to perform the following queries:

Query 1: Returns the same result set as this SELECT statement, but don't use a join. Instead, use a subquery in a WHERE clause that uses the IN keyword.

```
SELECT DISTINCT category_name
FROM category c JOIN product p
  ON c.category_id = p.category_id
ORDER BY category_name;
```

Query 2: Answers this question: Which products have a list price that's greater than the average list price for all products?

Return the product_name and list_price columns for each product. Sort the results by the list_price column in descending sequence.

Query 3: Display the category_name column from the Categories table.
Return one row for each category that has never been assigned to any product in the Products table. To do that, use a subquery introduced with the NOT EXISTS operator.

Query 4: Display the three columns: email_address, order_id, and the order total for each customer. To do this, you can group the result set by the email_address and order_id columns. In addition, you must calculate the order total from the columns in the order_items table.

Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return two columns: the customer's email address and the largest order for that customer. To do this, you can group the result set by the email_address.

Sample Run:

```
+----------------------------+----------+------------------+
| email_address              | order_id | max_order_total |
+----------------------------+----------+------------------+
| allan.sherwood@yahoo.com   |        1 |         3306.29 |
| barryz@gmail.com           |        2 |         1671.69
```

Query 5: Display the name and discount percent of each product that has a unique discount percent. In other words, don't include products that have the same discount percent as another product.
Sort the results by the product_name column.

Query 6: Use a correlated subquery to return one row per customer, representing the customer's oldest order (the one with the earliest date). Each row should include these three columns: email_address, order_id, and order_date.

Sample Run:

```
+----------------------------+----------+---------------------+
| email_address              | order_id | order_date          |
+----------------------------+----------+---------------------+
| allan.sherwood@yahoo.com   |        1 | 2015-03-28 09:40:28 |
| barryz@gmail.com           |        2 | 2015-03-28 11:23:20 |
```

Query 7: For each product whose items were sold in more than one sales transaction, display the product ID, product name, and list price.

Query 8: Display the last name and first name of customers who have purchased an item that costs more than $300. Use a subquery.

Sample Run:

```
+-----------+------------+
| Last_Name | First_Name |
+-----------+------------+
| Sherwood  | Allan      |
| Zimmer    | Barry      |
```

Query 9: Display the last name, first name, and email address of the customers who made the purchase with order IDs 1, 2, and 3. Use a subquery.

Query 10: Display the last name, first name, and email address of customers who have purchased an item that was supplied by a vendor with a Company Name that begins with the letter H. Use a subquery.

Sample Run:

```
+-----------+------------+-----------------------------+
| Last_Name | First_Name | email_address               |
+-----------+------------+-----------------------------+
| Zimmer    | Barry      | barryz@gmail.com            |
| Brown     | Christine  | christineb@solarone.com     |
```

Query 11: For each product that has more than two items sold within all sales transactions, retrieve the product id, product name, and product price.

Sample Run:

```
+------------+----------------------+------------+
| product_id | product_name         | list_price |
+------------+----------------------+------------+
|       1234 | Dell XPS 8700        |     999.00 |
|       2234 | Lenovo H50 Desktop   |    1199.00
```

Submission

- You will need to label your assignment with your first initial, last name, and the name of the assignment.
- Zip the files to upload to Insight (yourname_assignment9.zip).
- Submit the zipped file containing the script and output TXT via Insight.
- Remember to include the query number as a comment at each step.
- Read your output TXT file before you turn it in.