Assignment Chapter 7

6 points for question 1.

7 points for questions 2, 3.

10 points for questions 4, 5, 6.

Total 50 points

1. Write a SELECT statement that 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 categories c JOIN products p

ON c.category\_id = p.category\_id

ORDER BY category\_name

1. Write a SELECT statement that 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. Should return 2 rows.

1. Write a SELECT statement that returns 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.

Hint: running the following 2 queries should help check your answer

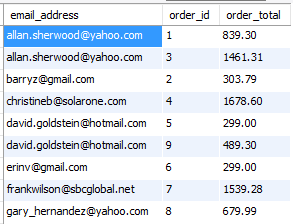
SELECT DISTINCT category\_id, category\_name FROM categories ORDER BY category\_id;

SELECT DISTINCT category\_id FROM products;

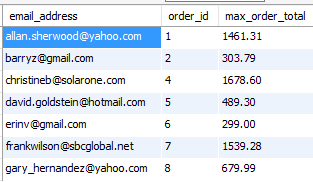
1. Write a SELECT statement that returns three columns for each customer:

* email\_address
* order\_id
* SUM((item\_price - discount\_amount) \* quantity) AS order\_total

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. This will not require any subqueries, and it should produce the result below.



Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return three columns: the customer’s email address, order\_id and the largest order for that customer. To do this, you can group the result set by the email\_address. This should return the following table



1. Write a SELECT statement that returns 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.

Hint: see “better question 6” in ExerciseSolutionsChapter7.html in Canvas.

1. 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.

Hint: see “question 7” in ExerciseSolutionsChapter7.html in Canvas.