CSCD 327 Lab #5 (15 points)
Due: July 30, 2014

Write the following queries in SQL using *database_3*. Please include both the queries and the query results in your submission.

1. List the book tittle and retail price for all books with a retail price lower than the average retail price of all books sold by the bookstore.
2. Determine which books cost less than the average cost of books in the same category. List the title, the category, and the cost of these books.
3. Determine which orders were shipped to the same state as order 1014.
4. Determine which orders had a higher total order amount than order 1002. [Note: total order amount = sum(Quantity * PaidEach)].
5. List the title of all books in the same categories as books previously purchased by customer 1007. Don't include books this customer has already purchased.
6. List the shipping city and state for the order that had the longest shipping delay. [Note: shipping delay = **datediff**(shipdate, orderdate)].
7. Determine which customers placed orders for the least expensive book (in terms of regular retail price) carried by the bookstore.
8. Determine the number of different customers who have placed an order for books written or co-written by James Austin.

Write the following queries in SQL using *database_4*. Please include the queries and the query results in your submission.

9. Find the instructor earning the highest salary.
10. Find the sections that had the maximum enrollment in Spring 2010. [Note: Enrollment is the number of students taking a course section.]
11. Find the lowest, across all departments, of the per-department maximum salary, along with the department name.

Write the following queries in SQL using *database_5*. Please include the queries and the query results in your submission.

12. 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.

13. 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.

14. Write a SELECT statement that returns three columns: email_address, order_id, and the order amount for each customer each order. To do this, you can group the result set by the email_address and order_id columns. The order amount is *sum((item_price – discount_amount) * quantity)* from 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 amount for that customer. To do this, you can group the result set by the email_address.

15. 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.