# ISOM 671: Managing Big Data (Assignment 1)

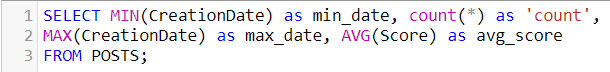
Qiyu Wang

qiyu.wang@emory.edu

**There are 20 numbered questions. Please answer them all and submit your assignment as a single PDF or Word file by uploading it to course canvas page. You should provide: SQL statements, results of the SQL statement (typically copy first 10 rows), and answers to questions, if any.**

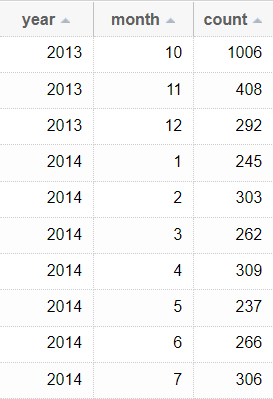
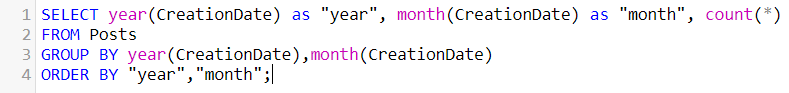
**Pets Stackexchange (**[**http://pets.stackexchange.com/**](http://pets.stackexchange.com/)**) is a Q&A forum for pets. This assignment is based on data of this forum. As other Stackexchange-branded forums, the pet exchange lists questions by the number of votes, answers, and views. Questions can be tagged so that users can easily explore related questions. You can use the StackExchange data explorer (****http://data.stackexchange.com/) to explore the data. For the first five questions, please use the data explorer for the Pets Stackexchange (**[**http://data.stackexchange.com/pets/query/new**](http://data.stackexchange.com/pets/query/new)**). Alternatively, if you prefer, you can download a data dump and import the database to your local installation of MySQL.**

1. Using the posts table, find out the number of posts, the minimum creation date (as min\_date), the maximum creation date (as max\_date), and average score (as avg\_score).

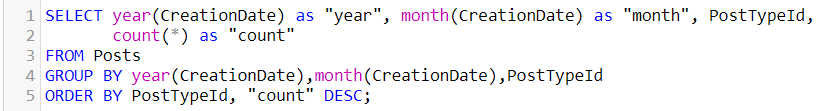


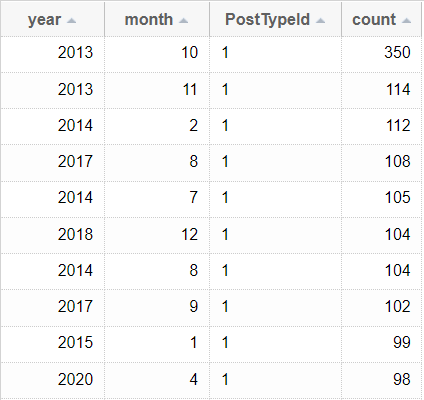


2. We want to get some ideas of how many posts were written each month. Use SQL to count the number of posts by year-month. Note that by year-month, we mean that May 2013 and May 2014 should be considered as different year-months. The result table should show year-month and count and order results by year-month.

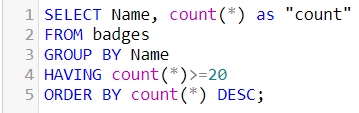
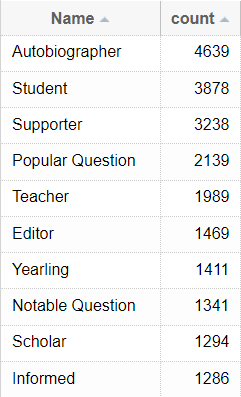


3. We know that there are different types of posts, as reflected by the PostTypeID: the original posts (i.e., 1), follow up posts (i.e., 2), survey questions (i.e., 4), and unknown (i.e., 5). Please use SQL to get the number of posts by year-month and by post type. The results table should show: year, month, posttypeid, and count (label: cnt). Use this result to answer the following question: which year-month has the most original posts?

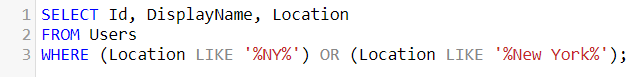
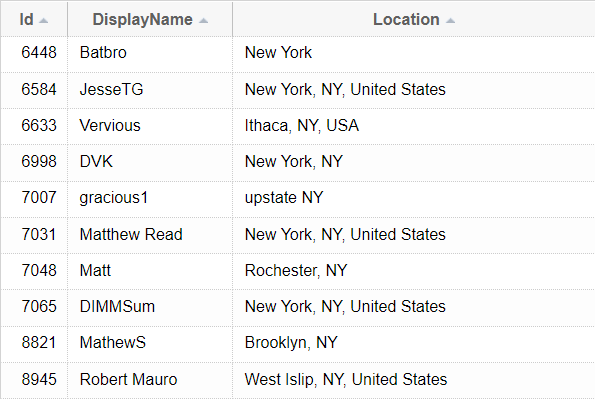


The October 2013 has the most original post.

4. Popular badges. Pets Exchange has an elaborate badging system. Use the badges table to find the most common type of badges. Specifically, show badge name (label name) and the number of people who won it (label cnt), limiting results to badges with at least 20 winners and sort the results by cnt in a descending order.

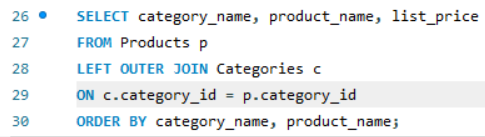
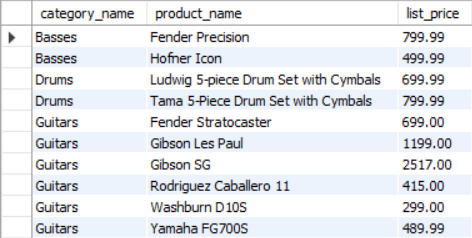
5. (use the users table) Find the number of users who report being located in the New York state. These include people who report themselves in New York city or in the NY state. Be as accurate as possible as the self-reported location may take different forms. Report id, displayname, and location in your result set.

**Please use the my\_guitar\_shop database provided on canvas to explore the data in your local MySQL instance and answer the next questions.**

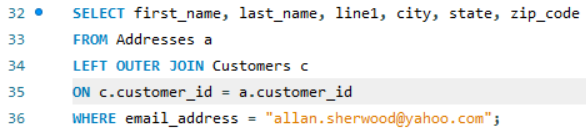
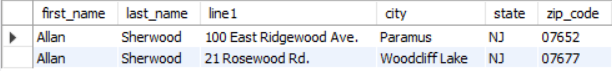
6. Write a SELECT statement that joins the Categories table to the Products table and returns these columns: category\_name, product\_name, list\_price.

Sort the result set by category\_name and then by product\_name in ascending sequence.

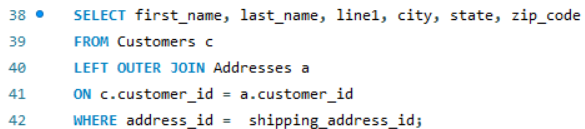
7. Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: first\_name, last\_name, line1, city, state, zip\_code.

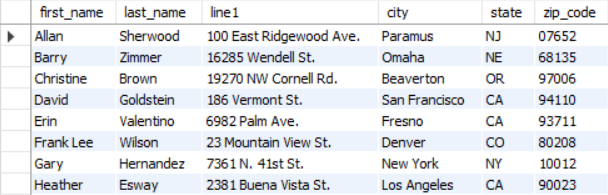
Return one row for each address for the customer with an email address of allan.sherwood@yahoo.com.

8. Write a SELECT statement that joins the Customers table to the Addresses table and returns these columns: first\_name, last\_name, line1, city, state, zip\_code.

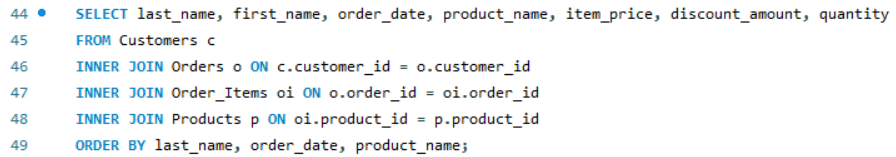
Return one row for each customer, but only return addresses that are the shipping address for a customer.

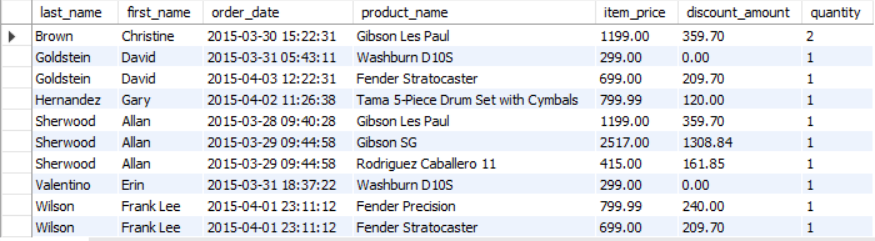




9. Write a SELECT statement that joins the Customers, Orders, Order\_Items, and Products tables. This statement should return these columns: last\_name, first\_name, order\_date, product\_name, item\_price, discount\_amount, and quantity.

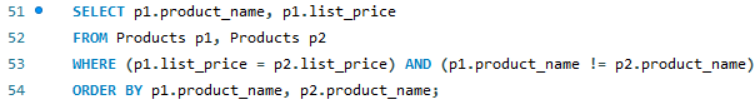
Use aliases for the tables. Sort the final result set by last\_name, order\_date, and product\_name.

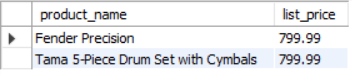




10. Write a SELECT statement that returns the product\_name and list\_price columns from the Products table.

Return one row for each product that has the same list price as another product. Sort the result set by product\_name.

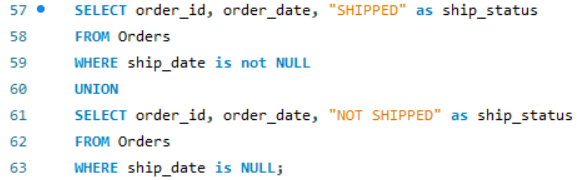


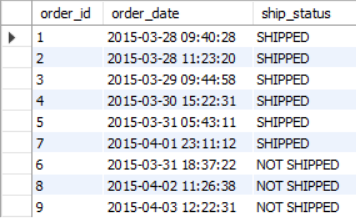


11. Use the UNION operator to generate a result set consisting of three columns from the Orders table:

* ship\_status A calculated column that contains a value of SHIPPED or NOT SHIPPED
* order\_id The order\_id column
* order\_date The order\_date column

If the order has a value in the ship\_date column, the ship\_status column should contain a value of SHIPPED. Otherwise, it should contain a value of NOT SHIPPED. Sort the final result set by order\_date.

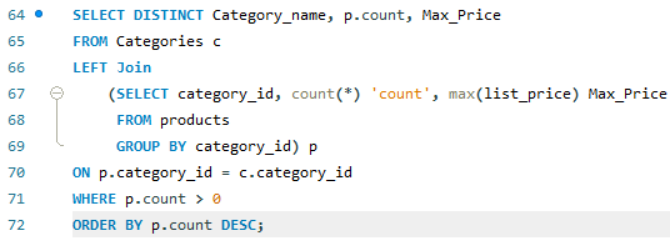
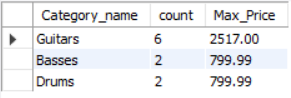




12. Write a SELECT statement that returns one row for each category that has products with these columns:

* The category\_name column from the Categories table
* The count of the products in the Products table
* The list price of the most expensive product in the Products table

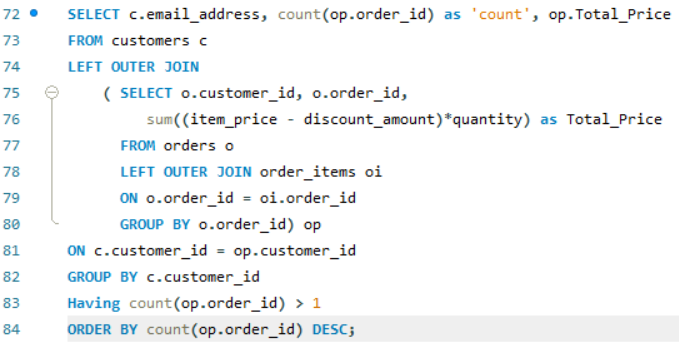
Sort the result set so the category with the most products appears first.

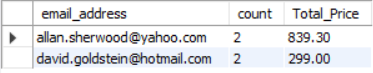
 

13. Write a SELECT statement that returns one row for each customer that has orders with these columns:

* The email\_address from the Customers table
* A count of the number of orders
* The total amount for each order (Hint: First, subtract the discount amount from the price. Then, multiply by the quantity.)

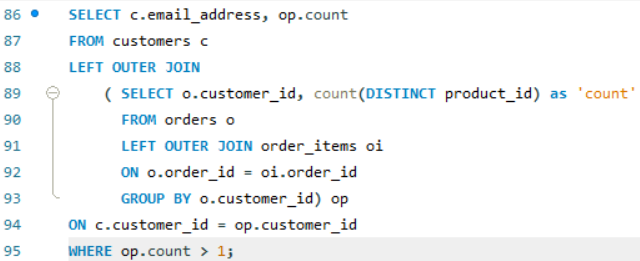
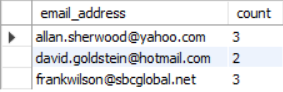
Return only those rows where the customer has more than 1 order. Sort the result set in descending sequence by the sum of the line item amounts.





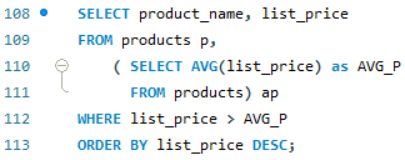
14. Write a SELECT statement that answers this question: Which customers have ordered more than one product? Return these columns:

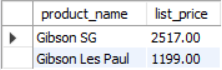
* The email address from the Customers table
* The count of distinct products from the customer’s orders

15. 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?

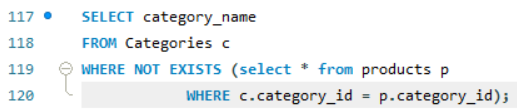
Return the product\_name and list\_price columns for each product. Sort the results by the list\_price column in descending sequence.



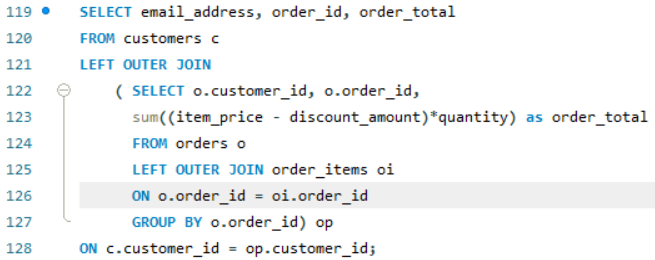
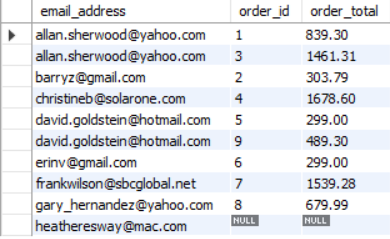


16. Write a SELECT statement that returns the category\_name column from the Categories table.

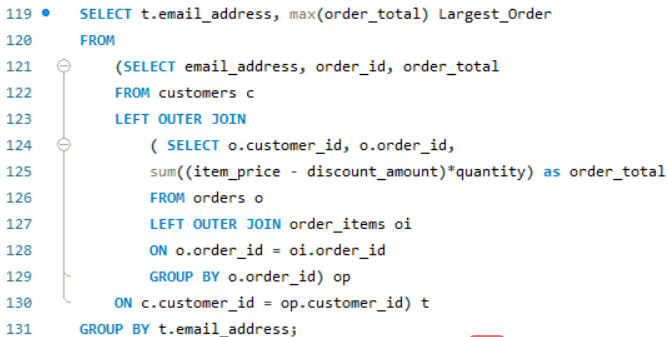
Return one row for each category that has never been assigned to any product in the Products table. To do that, use a subquery introduced with the NOT EXISTS operator.

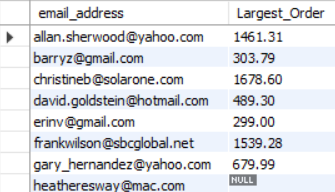
 

17. Write a SELECT statement that returns three columns: email\_address, order\_id, and the order total for each customer and order; you must calculate the order total from the columns in the Order\_Items table.

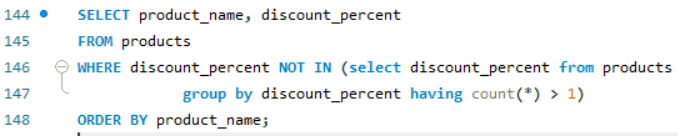
 

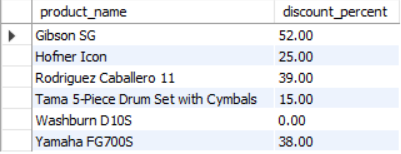
Write a second SELECT statement that uses the first SELECT statement in its FROM clause. The main query should return two columns: the customer’s email address and the largest order for that customer.





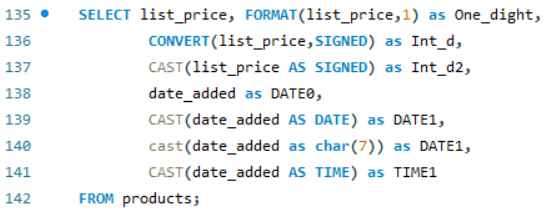
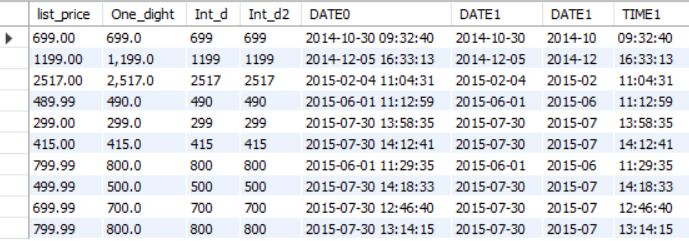
18. Write a SELECT statement that returns the name and discount percent of each product that has a unique discount percent. In other words, don’t include products that have the same discount percent as another product. Sort the results by the product\_name column.





19. Write a SELECT statement that returns these columns from the Products table:

* The list\_price column
* A column that uses the FORMAT function to return the list\_price column with 1 digit to the right of the decimal point
* A column that uses the CONVERT function to return the list\_price column as an integer
* A column that uses the CAST function to return the list\_price column as an integer
* The date\_added column
* A column that uses the CAST function to return the date\_added column with its date only (year, month, and day)
* A column that uses the CAST function to return the date\_added column with just the year and the month
* A column that uses the CAST function to return the date\_added column with its full time only (hour, minutes, and seconds)

20. Write a SELECT statement that returns these columns from the Orders table:

* The card\_number column
* The length of the card\_number column
* The last four digits of the card\_number column
* A column that displays the last four digits of the card\_number column in this format: XXXX-XXXX-XXXX-1234. In other words, use Xs for the first 12 digits of the card number and actual numbers for the last four digits of the number.

