# The University of Georgia
# Department of MIS
**MIST 4610**
Final Examination
Spring 2016

**By submitting this exam you acknowledge that the completed exam is entirely your own work and you have complied with the Academic Honesty program of the University of Georgia.**

**Submitted exams are analyzed to identify similarities in content and structure.**

**You may consult your notes and textbook and use your computer for testing SQL queries or drawing data models. You may not consult anyone else. You must be physically in the classroom.**

**Please submit this exam to the specified eLC dropbox as a single PDF.**

## Answers

Use the ClassicModels database, the data model is on the book web site, to answer the following requests using SQL.

## Short eLC Quiz (5 points)

## SQL 1 (15 points - 5 each)

1. List the names of sales representatives and their email address. Sort by last name.

```
SELECT firstName, lastName, email FROM Employees
WHERE jobTitle = "Sales Rep"
ORDER BY lastName;
```

2. List the names of products that have 'window' in their product description.

```
SELECT productName FROM Products
WHERE productDescription REGEXP '[Ww]indow' ;
```

3. How many orders where received in each month of 2004. Order by month ascending

```
SELECT MONTH(orderDate) AS MONTH, count(*) AS `Orders` FROM
    Orders
WHERE YEAR(orderDate) = 2004
GROUP BY MONTH
ORDER BY MONTH ASC;
```

## SQL 2 (40 points - 10 each)

1. Create a report listing the number of customers for each sales representative. Order by the number of customers descending

```
SELECT firstName, lastName, count(*) AS `Customers` FROM
    Customers, Employees
WHERE salesRepEmployeeNumber = employeeNumber
GROUP BY firstName, lastName
ORDER BY `Customers` DESC;
```

2. What is the profitability of each product line, where profit is the difference between the selling and buying price of a product.

```
SELECT productLine, FORMAT(SUM(quantityOrdered*(priceEach -
    buyPrice)),0) AS Profit
FROM Products, Orders, OrderDetails
WHERE Products.productCode = OrderDetails.productCode
AND OrderDetails.orderNumber = Orders.orderNumber
GROUP BY productLine;
```

3. What is the percentage of each month's payments for 2004 compared to the year's total payments for 2004.

```
SELECT MONTH(paymentDate)
AS Month, round(sum(amount)*100/(SELECT sum(amount) FROM Payments
    WHERE YEAR(paymentDate) = 2004 ),2) AS Percent FROM Payments
    WHERE YEAR(paymentDate) = 2004
    GROUP BY MONTH(paymentDate);
```

4. Write a procedure to change the MSRP of a given product line by a given percentage. Test with Ships and a decrease of 5%.

```
DELIMITER //
CREATE PROCEDURE changePrice (
IN pLine CHARACTER(20),
IN priceChange DECIMAL(5,2)
LANGUAGE SQL
BEGIN
UPDATE Products
SET MSRP = SMRP*(1 + priceChange)
WHERE productLine = pLine;
END //
```

## R (40 points - 10 each)

1. ClassicModels is considering creating a single consolidated warehouse to serve its US customers and wants your advice on a suitable location. As a first step, calculate the

average location (average latitude and longitude) of each of the US customers and show it on a Google map.

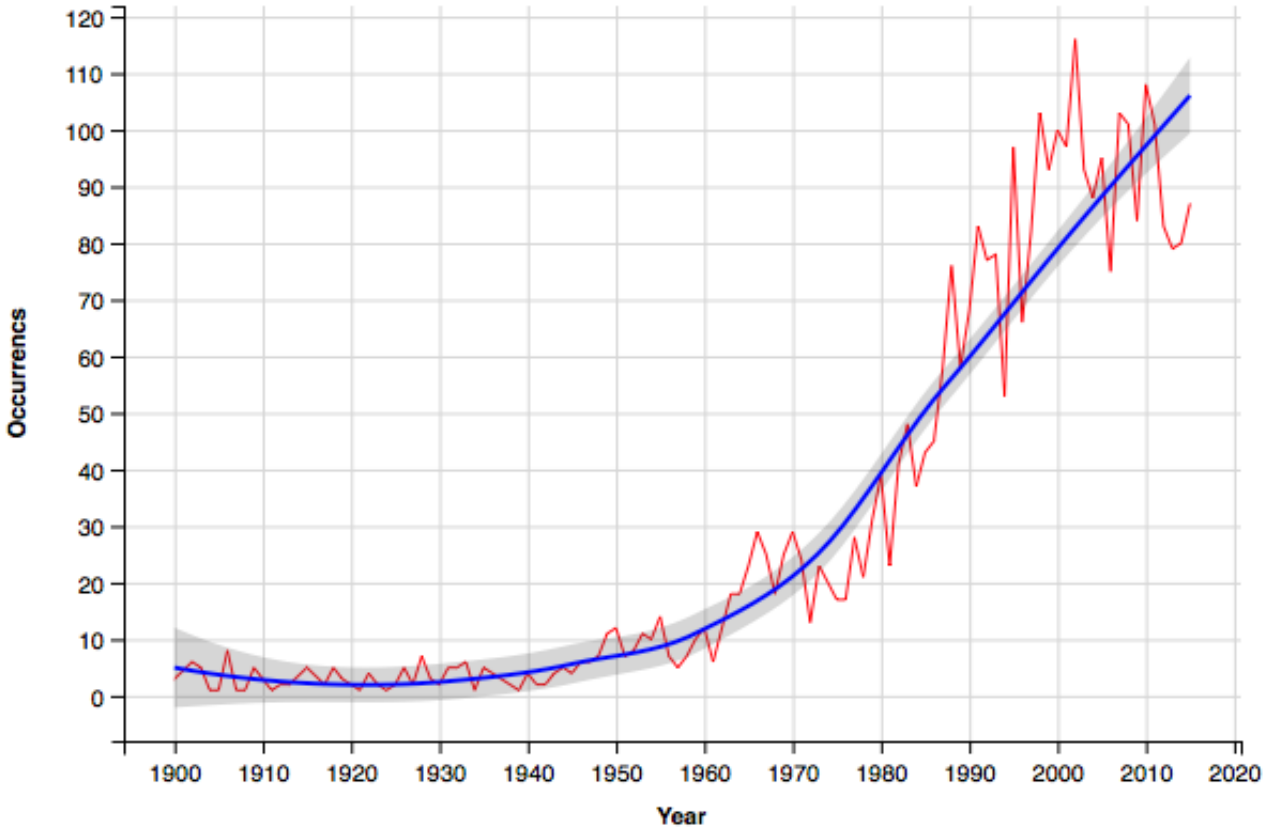
```
library(ggplot2)
library(ggmap)
library(mapproj)
library(readr)
library(DBI)
# connect to the database
conn <- dbConnect(RMySQL::MySQL(),"wallaby.terry.uga.edu", dbname
    = "ClassicModels", user = "student", password = "student")
m <-    dbGetQuery(conn,"SELECT avg(Y(customerLocation)) as
    Longitude, avg(X(customerLocation)) as Latitude from
    Customers where country = 'USA';")
map <- get_googlemap('united states',markers=m,zoom=4)
ggmap(map)
```

2. You have access to a dataset <http://people.terry.uga.edu/rwatson/data/AmericasDisasters.csv> containing details of natural disasters in the Americas.

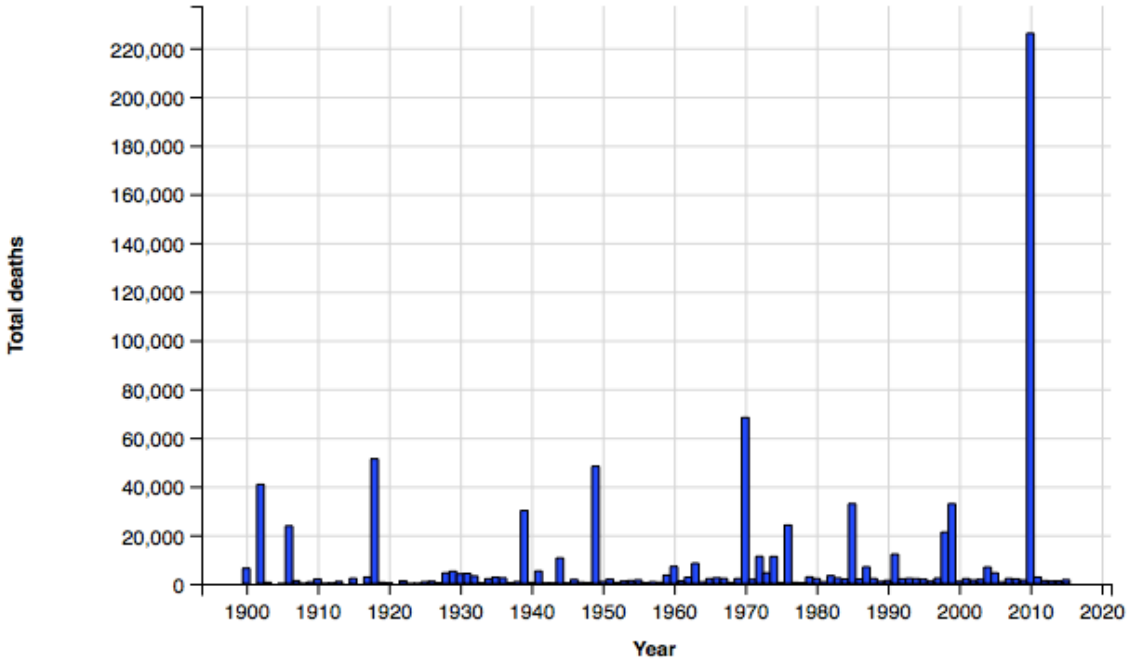
   a. Create a line graph with a smoother of the number of occurrences. Provide appropriate axes, labels, and formatting.

   b. Create a bar graph for the total number of deaths. Provide appropriate axes, labels, and formatting.

```
library(ggvis)
library(readr)
library(sqldf)
options(sqldf.driver = "SQLite")
url  <-    'http://people.terry.uga.edu/rwatson/data/
    AmericasDisasters.csv'
t <-  read_delim(url, delim=",")
t %>% ggvis(~year,~occurrence) %>% layer_lines(stroke:='red') %>%
  layer_smooths(se=T,stroke:='blue') %>%
  add_axis('y', title = "Occurrencs", title_offset=50) %>%
  add_axis('x', title ='Year', format='####')
```




```
library(ggvis)
library(readr)
url  <-    'http://people.terry.uga.edu/rwatson/data/
    AmericasDisasters.csv'
t <-  read_delim(url, delim=",")
t %>% ggvis(~year,~Total_deaths) %>% layer_bars(fill:='blue') %>%
  add_axis('y', title = "Total deaths", title_offset=100) %>%
  add_axis('x', title ='Year', format='####')
```




3. The file http://people.terry.uga.edu/rwatson/data/CreditCardDefaults.csv contains 30,000 records of credit card defaults in Taiwan.[1] For each level of education, compute using MapReduce the maximum, average, and minimum credit limit balance. Note that the amounts are in the Taiwanese currency, and the exchange rate is about .03 USD.

```
library(readr)
library(rmr2)
library(reshape)
rmr.options(backend = "local") # local or hadoop
url  <-  'http://people.terry.uga.edu/rwatson/data/
    CreditCardDefaults.csv'
t <- read_delim(url, delim = "," )
#Save credit details in hdfs file
hdfs.credit <- to.dfs(data.frame(t))
#Mapper for computing amounts for each gender
mapper <- function(k,v) {
  key <- v$EDUCATION
  value <- v$LIMIT_BAL
  keyval(key,value)
}
#Reducer to report the results
reducer <- function(k,v) {
  key <- k #origin
  value <- c(max(v), round(mean(v),0), min(v)) #Vector
```

[1] Source: https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients

```
  keyval(key,value) }
out = mapreduce(input = hdfs.credit, map = mapper, reduce =
    reducer)
df1 = as.data.frame(from.dfs(out))
df1$measure <- c('Max','Mean','Min')
#Reshape data
r2 <- cast(df1,key ~ measure, value="val")
colnames(r2)[1] = 'Education'
head(r2) #Show results
```

```
  Education     Max   Mean   Min
1         0  360000 217143 50000
2         1 1000000 212956 10000
3         2  800000 147062 10000
4         3  760000 126550 10000
5         4  710000 220894 20000
6         5  550000 168164 10000
```

4. The web site http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_statistics/html/table_01_15.html_mfd contains details of annual US motor vehicle production. Download the Excel or CSV version of table 1-15, clean it up, and convert it into a form suitable for use with R. Compute the average number of cars produced each year.

```
library(readr)
library(reshape)
library(sqldf)
options(sqldf.driver = "SQLite")
url <- "car_production.csv"
r <- read_delim(url,delim = ',')
colnames(r) <- c('measure', 1960,1965,1970,1975,1980,1985,1990:2014)
m1 <- melt(as.data.frame(r),id='measure')
colnames(m1) <- c('measure','year','count')
head(m1)
sqldf('select avg(count)*1000 from m1 where measure = "Production, total";')
```

10533.77