## SQL Challenge

Introduction

In this case study, we are asking you to work with a publicly available dataset, and create the reports listed below and answer the questions.

Database: Dell DVD Store Database (<http://linux.dell.com/dvdstore/>)

Please see the link below for more information about the database: (<https://dataedo.com/samples/html/Dell_DVD_Store/doc/Dell_DVD_Store_12/modules/Dell_DVD_Store_database_diagram_107/module.html>)

Assignment

1. Create a report that provides the following columns:

|  |  |
| --- | --- |
| orderid |  |
| orderdate |  |
| customerid |  |
| totalamount |  |
| order quantity |  |
| category name |  |
| title |  |
| city |  |
| state |  |
| country |  |
| age |  |
| income |  |
| gender |  |
| customer type | “new” if this is the first order of the customer. “existing” if the customer placed at least one order before the current order |
| # of previous orders | Count of previous orders placed by the customer |
| Total $ amount of previous orders | Total amount of previous orders placed by the customer |
| Order fulfilled | “full”: if the full quantity ordered is available, “partial”: if quantity available is greater than 0 but less than the order quantity, “backorder”: if order cannot be fulfilled |
| Expected order fulfillment date | If order cannot be fulfilled fully, the date the remaining quantity will be available |

1. By using the report created above, please create summary tables to answer the following questions?
   1. What percent of the sales come from new vs. existing customers?
   2. What percent of orders were fulfilled (partial, full, backorder)?
   3. What is the distribution of sales by category?
   4. What is the monthly revenue and what is the cumulative revenue as of each month within a year (cumulative revenue resets at the beginning of each year)?
   5. What is the distribution of sales by customer age buckets?
2. Consider the following R code: ` cumsum(rnorm(100)) `. How would you describe the data being generated?

* The rnorm(100) function generates 100 random numbers the cumsum function wrapped around it then cumulatively sums all rndom generated values from the x=rnorm (100)
* vairable

1. Describe in your own words the following R code chunk and qualitatively characterize the resulting data.

* the initial lines are defining the variables N, inds, b, a, s, and x
* the rnorm function that s is set to isn’t locked in with a seed so it effectively generates 512 new numbers everytime the code is executed
* x is then set to the fast fourier transform of that series of randomly generated values and converts the random series into a frequency
* the time seriess function plots the standardized ‘y’ data since ‘y’ returns a non- normalized transform of the series of values

|  |
| --- |
| N <- 512  inds <- c(8,16)  b <- 30  a <- 128  s <- rnorm(N)  x <- fft(s)  logind <- rep(FALSE, N/2-1)  logind[inds] <- TRUE  logind <- c(FALSE, logind, FALSE, rev(logind))  x[logind] <- b \* x[logind]  y <- Re(fft(x, inverse = TRUE) / length(x))  result <- ts(y / sd(y), frequency = a) |

1. Write the R code that builds a model to predict "c", characterize its accuracy, and describe the steps you took as well as its accuracy.

|  |
| --- |
| df <- data.frame(a = c(rnorm(80), rnorm(20,1,2))  , b = c(rnorm(80,0,3), rnorm(20,2,2))  , c = c(rep(TRUE,80), rep(FALSE,20))) |

Deliverables:

Please provide the SQL code as well as the results for each question. If the report for #1 is large, a subset of the report is acceptable. It is possible that some of the information requested cannot be produced with the data available. In that case, please provide an explanation why it cannot be produced. Use window functions where applicable.

Wrap Up

We wish you the best of luck with this case study. If you have any questions, please do not hesitate to reach out.