Procedimento de extração TDB Produção

José Cassio dos Santos Jr.

8 de julho de 2015

Este script descreve em detalhe o código fonte que executa a extração. Abaixo seguem partes do código fonte comentados, na ordem em que são executados:

# ATENÇÃO: mudar data de processamento abaixo  
dataproc <- "20150630"

# source scripts necessários  
source("./R/f\_leCsvCARPROD\_raw.R")  
source("./R/f\_SICLID\_raw.R")  
source("./R/f\_consignado\_raw.R")

# verifica e prepara ambiente de libraries necessárias  
if (!requireNamespace("openxlsx", quietly = TRUE)) {  
 stop("openxlsx needed for this function to work. Please install it.",  
 call. = FALSE)  
}   
if(!require(openxlsx)){install.packages("openxlsx")}  
  
if (!requireNamespace("dplyr", quietly = TRUE)) {  
 stop("dplyr needed for this function to work. Please install it.",  
 call. = FALSE)  
}  
if(!require(RODBC)){install.packages("RODBC")}  
if (!requireNamespace("dplyr", quietly = TRUE)) {  
 stop("dplyr needed for this function to work. Please install it.",  
 call. = FALSE)  
}  
if(!require(RODBC)){install.packages("RODBC")}

# constantes usadas  
# constantes  
rawDir <- "./rawdata" # pasta com arquivos brutos (a partir da pasta de trabalho)  
tidyDir <- "./tidydata" # pasta com arquivos processados (a partir da pasta de trabalho)  
dtproc <- paste0(substr(dataproc,5,6),".",substr(dataproc,1,4))

# checa se diretorio corrente raw já carregado para o mês  
if (!file.exists(paste0(rawDir,"/",dtproc))){  
 stop("Diretório com arquivos origem do mês não está presente.",  
 call. = FALSE)  
}

# criar diretório do mês no diretório tidydata, caso não exista  
if (!file.exists(paste0(tidyDir,"/",dtproc))){  
 dir.create(file.path(tidyDir, dtproc))  
}

# chama função para criar os arquivos tidy para alimentar os dados de Produção Cartão   
# da planilha TDB Produção, a partir dos arquivos gerados pelo SAS  
f\_SICLID\_raw(dataproc)

# chama função para criar os arquivos tidy para alimentar os dados de Consignado   
# da planilha TDB Produção, a partir dos arquivos gerados pelo SAS  
f\_consignado\_raw(dataproc)

### Função f\_SICLID\_raw

# constates  
 mm\_aaaa <- paste0(substr(dataproc,5,6),".",substr(dataproc,1,4))  
 aaaamm <- paste0(substr(dataproc,1,4),substr(dataproc,5,6))  
 carprod\_in <- paste0("./rawdata/", mm\_aaaa, "/CARPROD\_", dataproc,".txt")  
 carteprod\_in <- paste0("./rawdata/", mm\_aaaa, "/carte\_prod\_TDB\_PRODUCAO\_",dataproc,".csv")  
 carteseguros\_in <- paste0("./rawdata/", mm\_aaaa, "/carte\_seguros\_", aaaamm,".csv")  
 cartecartoes\_in <- paste0("./rawdata/", mm\_aaaa, "/carte\_cartoes\_", aaaamm,".csv")  
 carteamort\_in <- paste0("./rawdata/", mm\_aaaa, "/carte\_amort\_",aaaamm,".csv")  
 fileout1 <- paste("./tidydata", mm\_aaaa, "tidyCarProd.csv", sep = "/")  
 fileout2 <- paste("./tidydata", mm\_aaaa, "tidyCarteProd.csv", sep = "/")  
 fileout3 <- paste("./tidydata", mm\_aaaa, "tidyCarteSeguros.csv", sep = "/")  
 fileout4 <- paste("./tidydata", mm\_aaaa, "tidyCarteCartoes.csv", sep = "/")  
 fileout5 <- paste("./tidydata", mm\_aaaa, "tidyCarteAmort.csv", sep = "/")

################################  
 # aba <mês>", DADOS DE PRODUÇÃO CREDIÁRIO  
 ################################  
 # ler arquivo de janeiro/2015 a maio/2015  
 df\_creprod <- f\_leCsvCARPROD\_raw(carprod\_in, !temHeader)  
 # mudando nomes  
 new\_names <- c("SOCIEDADE","DATAPROC","CHAINEORI","PRODALP","MODPAY","TIPOTNC","NBFI","MTFI","DURFIN","TAC", "TEC", "TXCLM", "TXVDRFINAL", "SEGURO", "RETENCAO")  
 names(df\_creprod) <- new\_names  
 # rearranjando colunas  
 df\_creprod <-   
 df\_creprod %>%  
 select(SOCIEDADE, CHAINEORI,PRODALP,MODPAY,TIPOTNC,NBFI,MTFI,  
 DURFIN,TAC, TEC, TXCLM, TXVDRFINAL, SEGURO, RETENCAO, DATAPROC)  
   
 # colocar filtro para gravar somente em baseCarProd o mês corrente  
 # (pois não pode alterar valores dos meses anteriores.   
 #-------------OBS  
 # orientação é forçar os meses anteriores como o mes corrente para bater com relatório  
 # atual. Depois que alinhar como proceder, devemos alterar este procedimento  
 df\_creprod <-  
 df\_creprod %>%  
 mutate(DATAPROC = paste0(substr(mm\_aaaa,4,7),substr(mm\_aaaa,1,2)))

###################  
 # aba <mês>, DADOS DE PRODUÇÃO CARTÃO  
 ###################  
 # ler arquivos de janeiro/2015 a maio/2015  
 df\_carprod <- f\_leCsvCARPROD\_raw(carteprod\_in,temHeader)  
 # eliminando coluna COUNT\_OF\_SOCIEDADE  
 df\_carprod <-   
 df\_carprod %>%  
 select (-(COUNT\_OF\_SOCIEDADE))  
 #-------------OBS  
 # orientação é forçar os meses anteriores como o mes corrente para bater com relatório  
 # atual. Depois que alinhar como proceder, devemos alterar este procedimento  
 df\_carprod <-  
 df\_carprod %>%  
 mutate(DATAPROC = paste0(substr(mm\_aaaa,4,7),substr(mm\_aaaa,1,2)))

####################  
 # aba <mês>, DADOS DE ADESÃO E BASE DE SEGUROS  
 ##################  
 # ler arquivos de janeiro/2015 a maio/2015  
 df\_seguro <- f\_leCsvCARPROD\_raw(carteseguros\_in,temHeader)  
 #-------------OBS  
 # orientação é forçar os meses anteriores como o mes corrente para bater com relatório  
 # atual. Depois que alinhar como proceder, devemos alterar este procedimento  
 df\_seguro <-  
 df\_seguro %>%  
 mutate(DATAPROC = paste0(substr(mm\_aaaa,4,7),substr(mm\_aaaa,1,2)))

###################  
 # aba <mês>, DADOS DE CARTÃO  
 ########################  
 # ler arquivos de janeiro/2015 a maio/2015  
 df\_cartecartoes <- f\_leCsvCARPROD\_raw(cartecartoes\_in,temHeader)  
 #-------------OBS  
 # orientação é forçar os meses anteriores como o mes corrente para bater com relatório  
 # atual. Depois que alinhar como proceder, devemos alterar este procedimento  
 df\_cartecartoes <-  
 df\_cartecartoes %>%  
 mutate(DATAPROC = paste0(substr(mm\_aaaa,4,7),substr(mm\_aaaa,1,2)))

##################  
 # aba DadosAmort, DADOS DE AMORTIZACAO  
 #####################  
 # ler arquivos de janeiro/2015 a maio/2015  
 df\_carteamort <- f\_leCsvCARPROD\_raw(carteamort\_in,temHeader)  
 #-------------OBS  
 # orientação é forçar os meses anteriores como o mes corrente para bater com relatório  
 # atual. Depois que alinhar como proceder, devemos alterar este procedimento  
 df\_carteamort <-  
 df\_carteamort %>%  
 mutate(DATAPROC = paste0(substr(mm\_aaaa,4,7),substr(mm\_aaaa,1,2)))

# gravando os arquivos .csv  
 write.csv2(df\_creprod, file = fileout1)  
 write.csv2(df\_carprod, file = fileout2)  
 write.csv2(df\_seguro, file = fileout3)  
 write.csv2(df\_cartecartoes, file = fileout4)  
 write.csv2(df\_carteamort, file = fileout5)

### Função f\_consignado\_raw

# caminhos e arquivos  
 mm\_aaaa <- paste0(substr(anoMesDia,5,6),".",substr(anoMesDia,1,4))  
 rawDir <- "./rawdata"  
 tidyDir <- "./tidydata"  
 fileout\_cuboprod <- paste0(tidyDir,"/", mm\_aaaa,"/","riscbgn\_cubo\_producao\_",anoMesDia, "\_tidy.csv")  
 fileout\_cuboliqu <- paste0(tidyDir,"/", mm\_aaaa,"/","riscbgn\_cubo\_liquidacao\_",anoMesDia, "\_tidy.csv")  
 fileout\_basecart <- paste0(tidyDir,"/", mm\_aaaa,"/","riscbgn\_base\_carteira\_",anoMesDia, "\_tidy.csv")  
   
 # conexao Oracle  
 caminho <- "DWCTLPRD"  
 userid <- "usr\_pbgn\_ltra"  
 passwd <- "usr\_pbgn\_ltra"

# ----------------------------------------------  
 # TABELA PRODUCAO  
 # ----------------------------------------------  
   
 # abre conexão com Oracle  
 channel <- odbcConnect(caminho,uid=userid, pwd=passwd, believeNRows=FALSE)  
   
 # cria tabela temp.producao a partir do Oracle, segundo pesquisa sas já existente  
 # em consignado.sas  
 # Descrição:  
 # seleciona a partir das tabelas de fatos de Operações Mensais e Operações Realizadas  
 # os valores de montante de saldos contábeis não ???  
 # Também obtém as seguinte dimensões para poder realizar as sumarizações e   
 # filtragens:  
 # tb\_dim\_grpo\_prmt (correspondente bancário)  
 # tb\_dim\_prdt (produto)  
 # tb\_dim\_epdr (empresa cliente?)  
 # tb\_dim\_flal (filial BGN)  
 # tb\_dim\_prmt (Parceiro?)  
 # tb\_dim\_crtr (tipo de consignado?)  
 # tb\_dim\_cnal\_vnda (canal de venda)  
 # Período selecionado:  
 # seleciona sempre para todas as datas bases acima de janeiro/2013  
   
 cSQL\_temp <- paste0("Select round(a11.dt\_base/100) as dt\_base,",  
 "round(a11.dt\_crga/100) as dt\_ref,",  
 "A12.DS\_CNAL\_VNDA,",  
 "A13.DS\_EPDR,",  
 "A14.DS\_FLAL,",  
 "A14.Cd\_Flal,",  
 " A15.ds\_prdt,",  
 "a16.DS\_PRMT,",  
 "A22.DS\_GRPO\_PRMT,",  
 "a18.ds\_crtr,",  
 "a18.cd\_crtr as CD\_CRTR\_TEMP,",  
 "count(A11.ID\_OPRC\_RLZD) as QTD\_PRODUCAO,",  
 "sum(MT\_OPRC/100) as VLR\_PRODUCAO\_BRUTA,",  
 "sum(MT\_LBRD/100) as VLR\_PRODUCAO\_BRUTA\_SEM\_TAX,",  
 "sum(MT\_IOC/100) as VLR\_IOF,",  
 "sum(MT\_PRCL/100) as VLR\_PARCELA,",  
 "sum(MT\_BRTO/100) as VLR\_FUTURO,",  
 "sum(Mt\_lqdo\_cmcl/100) as VLR\_PRODUCAO\_LIQUIDA,",  
 "sum(((A11.MT\_OPRC/100-A11.MT\_LQDO\_CMCL/100))) as VLR\_PRODUCAO\_REFIN,",  
 "sum(((MT\_LBRD/100)-(A11.MT\_OPRC/100-A11.MT\_LQDO\_CMCL/100))) as VLR\_PRODUCAO\_REFIN\_SEM\_TAX,",  
 "sum((MT\_OPRC\*QT\_PRCL)/100) as MT\_BRUTO\_PRAZO\_PRD,",  
 "sum((A11.MT\_OPRC\*A11.QT\_PRCL\*A11.VL\_TAXA\_AMES)/10000) as MT\_BRUTO\_PRAZO\_JUROS\_PRD",  
 " From USR\_PBGN\_LOAD.TB\_FAT\_OPRC\_RLZD A11,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_CNAL\_VNDA A12,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_EPDR A13,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_FLAL A14,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_PRDT A15,",  
 " usr\_pbgn\_load.tb\_dim\_prmt a16,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_GRNT\_RGNL A17,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_CRTR A18,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_GRNT A19,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_CRDN A20,",  
 " USR\_PBGN\_LOAD.TB\_DIM\_OPRD A21,",  
 " usr\_pbgn\_load.tb\_dim\_grpo\_prmt a22",  
 " Where A11.ID\_CNAL\_VNDA = A12.ID\_CNAL\_VNDA and",  
 " A11.ID\_EPDR = A13.ID\_EPDR and",  
 " A11.ID\_FLAL = A14.ID\_FLAL and",  
 " A11.ID\_PRMT = a16.ID\_PRMT and",  
 " A11.id\_prdt = A15.id\_prdt and",  
 " A11.id\_grnt\_rgnl = A17.id\_grnt\_rgnl and",  
 " A11.ID\_CRTR = A18.ID\_CRTR and",  
 " A11.id\_grnt = A19.id\_grnt and",  
 " A11.id\_crdn = A20.id\_crdn and",  
 " A11.id\_oprd = A21.id\_oprd and",  
 " A11.id\_grpo\_prmt = A22.ID\_GRPO\_PRMT and",  
 " round(a11.dt\_base/100) >= 201301 and",  
 " A11.DT\_BASE < to\_char(SYSDATE,'YYYY')\*10000 + to\_char(SYSDATE,'MM')\*100 + to\_char(SYSDATE,'DD')",  
 " Group By round(a11.dt\_base/100),",  
 "round(a11.dt\_crga/100),",  
 "A12.DS\_CNAL\_VNDA,",  
 "A13.DS\_EPDR,",  
 "A14.DS\_FLAL,",  
 "A14.Cd\_Flal,",  
 "A15.ds\_prdt,",  
 "a16.DS\_PRMT,",  
 "A22.DS\_GRPO\_PRMT,",  
 "a18.ds\_crtr,",  
 "a18.cd\_crtr;"  
 )  
  
 # executa a consulta  
 df\_temp\_producao <- sqlQuery(channel,cSQL\_temp, errors = TRUE)  
  
 # fecha conexão com Oracle  
 odbcClose(channel)  
 # força para todos os registros:  
 # condição de ATRASO = "Em Dia"  
 # FAIXA REPORT = "R0"  
 # acumula ocorrências de código de consignado fora dos grupos como geral 99-CONSIGNADO  
 df\_temp\_producao <-  
 df\_temp\_producao %>%  
 mutate (ATRASO = "Em Dia",  
 FAIXA\_REPORT = "R0",  
 id\_faix\_atrs = -1,  
 CD\_CRTR = ifelse(!(CD\_CRTR\_TEMP %in% c(50,52,53,55)), 99, CD\_CRTR\_TEMP))  
  
 # ------------ PASSO ADICIONAL PARA TEMP\_PRODUCAO  
 # troca vírgula por ponto em colunas selecionadas de data.frame  
 df\_temp\_producao <- data.frame(lapply(df\_temp\_producao, function(x) gsub(",", ".", x, fixed = TRUE)), stringsAsFactors = FALSE)  
  
 # ------------- PASSO 3 (OK) Valores conferem com SAS  
 # cria tabela riscbgn.cubo\_producao a partir da tabela temp.producao  
 # entrada df\_temp\_producao  
 # saida: df\_riscbgn\_cubo\_producao  
 # obs: para que a soma abaixo funcione é preciso gerar dataframe sem factors para  
 # valores numéricos e usar cláusula as.numeric para transformar de character para numerico!!!  
 df\_riscbgn\_cubo\_producao <-  
 df\_temp\_producao %>%  
 select (DT\_REF,DT\_BASE,DS\_CNAL\_VNDA,DS\_EPDR,DS\_FLAL,DS\_GRPO\_PRMT,  
 DS\_PRDT,DS\_CRTR,ATRASO,FAIXA\_REPORT,id\_faix\_atrs, QTD\_PRODUCAO,  
 VLR\_PRODUCAO\_BRUTA,VLR\_PRODUCAO\_BRUTA\_SEM\_TAX,VLR\_IOF,  
 VLR\_PARCELA,VLR\_FUTURO,VLR\_PRODUCAO\_LIQUIDA,VLR\_PRODUCAO\_REFIN,  
 VLR\_PRODUCAO\_REFIN\_SEM\_TAX,MT\_BRUTO\_PRAZO\_PRD,MT\_BRUTO\_PRAZO\_JUROS\_PRD) %>%  
 group\_by(DT\_REF,DT\_BASE,DS\_CNAL\_VNDA,DS\_EPDR,DS\_FLAL,DS\_GRPO\_PRMT,  
 DS\_PRDT,DS\_CRTR,ATRASO,FAIXA\_REPORT,id\_faix\_atrs) %>%  
 summarize(QTD\_PRODUCAO = sum(as.numeric(QTD\_PRODUCAO)),  
 VLR\_PRODUCAO\_BRUTA = sum(as.numeric(VLR\_PRODUCAO\_BRUTA)),  
 VLR\_PRODUCAO\_BRUTA\_SEM\_TAX = sum(as.numeric(VLR\_PRODUCAO\_BRUTA\_SEM\_TAX)),  
 VLR\_IOF = sum(as.numeric(VLR\_IOF)),  
 VLR\_PARCELA = sum(as.numeric(VLR\_PARCELA)),  
 VLR\_FUTURO = sum(as.numeric(VLR\_FUTURO)),  
 VLR\_PRODUCAO\_LIQUIDA = sum(as.numeric(VLR\_PRODUCAO\_LIQUIDA)),  
 VLR\_PRODUCAO\_REFIN = sum(as.numeric(VLR\_PRODUCAO\_REFIN)),  
 VLR\_PRODUCAO\_REFIN\_SEM\_TAX = sum(as.numeric(VLR\_PRODUCAO\_REFIN\_SEM\_TAX)),  
 MT\_BRUTO\_PRAZO\_PRD = sum(as.numeric(MT\_BRUTO\_PRAZO\_PRD)),  
 MT\_BRUTO\_PRAZO\_JUROS\_PRD = sum(as.numeric(MT\_BRUTO\_PRAZO\_JUROS\_PRD)))  
  
 # aqui gravar arquivo raw em riscbgn\_cubo\_producao\_AAAAMMDD\_raw.csv na pasta do mês de processamento  
 # este arquivo deve substituir os dados da aba base usada na planilha TDB Produção  
 write.csv2(df\_riscbgn\_cubo\_producao, file = fileout\_cuboprod)  
  
 # remove bases temporarias  
 rm(df\_temp\_producao, df\_riscbgn\_cubo\_producao)

# ----------------------------------------------  
 # TABELA LIQUIDACAO  
 # ----------------------------------------------  
   
 # entrada: tabelas oracle  
 # saida: df\_temp\_liquidacao  
 # cria tabela temp.producao  
  
 # abre conexao com oracle  
 channel <- odbcConnect(caminho,uid=userid, pwd=passwd, believeNRows=FALSE)  
  
 # cria tabela temp.liquidacao a partir do Oracle, segundo pesquisa sas já existente  
 # em consignado.sas  
 # Descrição:  
 # seleciona a partir das tabelas de fatos de Movimentações Financeiras Realizadas  
 # e Operações Realizadas os valores de montante de saldos contábeis não ???  
 # Também obtém as seguinte dimensões para poder realizar as sumarizações e   
 # filtragens:  
 # tb\_dim\_hist\_fncr (histórico financeiro)  
 # tb\_dim\_grpo\_prmt (correspondente bancário)  
 # tb\_dim\_prdt (produto)  
 # tb\_dim\_epdr (empresa cliente?)  
 # tb\_dim\_flal (filial BGN)  
 # tb\_dim\_prmt (Parceiro?)  
 # tb\_dim\_crtr (tipo de consignado?)  
 # tb\_dim\_cnal\_vnda (canal de venda)  
 # tb\_dim\_crdn (?)  
 # tb\_dim\_grnt\_rgnl (?)  
 # tb\_dim\_grnt (?)  
 # tb\_dim\_oprd (?)  
 # Período selecionado:  
 # seleciona sempre para todas as datas de carga acima de janeiro/2013  
 cSQL\_liq <- paste0("Select round(r.dt\_crga/100) as dt\_ref,", # -- Extrai ano e m?s (exemplo: 20140605 -> 201406)  
 "r.cd\_tipo\_mvmt,",  
 "x.cd\_hist\_fncr,",  
 "x.ds\_hist\_fncr,",  
 "a.cd\_cnal\_vnda,",  
 "a.ds\_cnal\_vnda,",  
 "g.ds\_oprd,",  
 "b.ds\_crdn,",  
 "d.ds\_grnt,",  
 "c.ds\_grnt\_rgnl,",  
 "e.ds\_epdr,",  
 "f.ds\_grpo\_prmt,",  
 "h.ds\_prdt,",  
 "i.ds\_prmt,",  
 "k.DS\_FLAL,",  
 "j.ds\_crtr,",  
 "j.cd\_crtr as cd\_crtr\_temp,",  
 "round(t.dt\_base/100) as dt\_base,", # -- Extrai ano e mês (exemplo: 20140605 -> 201406)  
 "(round(r.dt\_crga/10000) - round(t.dt\_base/10000)) \* 12 + mod(round(r.dt\_crga/100),100) - mod(round(t.dt\_base/100),100) as MOB,",  
 "sum(round(t.MT\_OPRC/100,2)) as Producao,",  
 "DECODE (x.cd\_hist\_fncr, '104',sum(round(r.vl\_mvmt/-100,2)),sum(round((r.vl\_mvmt\*t.qt\_prcl\*t.vl\_taxa\_ames)/ 10000,2))) as vl\_mvmt\_qtd\_tx,",  
 "DECODE (x.cd\_hist\_fncr, '104',sum(round((r.vl\_mvmt\*t.qt\_prcl\*t.vl\_taxa\_ames)/-10000,2)),sum(round(r.vl\_mvmt/ 100,2))) as vl\_mvmt,",  
 "DECODE (x.cd\_hist\_fncr, '104',sum(round((r.vl\_mvmt\*t.vl\_taxa\_ames)/-10000,2)),sum(round((r.vl\_mvmt\*t.vl\_taxa\_ames)/10000,2))) as vl\_mvmt\_tx",  
 " From usr\_pbgn\_load.tb\_fat\_mvmt\_fncr r,",  
 "usr\_pbgn\_load.tb\_dim\_hist\_fncr x,",  
 "usr\_pbgn\_load.Tb\_Fat\_Oprc\_Rlzd t,",  
 "usr\_pbgn\_load.tb\_dim\_cnal\_vnda a,",  
 "usr\_pbgn\_load.tb\_dim\_crdn b,",  
 "usr\_pbgn\_load.tb\_dim\_grnt\_rgnl c,",  
 "usr\_pbgn\_load.tb\_dim\_grnt d,",  
 "usr\_pbgn\_load.tb\_dim\_epdr e,",  
 "usr\_pbgn\_load.tb\_dim\_grpo\_prmt f,",  
 "usr\_pbgn\_load.tb\_dim\_oprd g,",  
 "usr\_pbgn\_load.Tb\_Dim\_Prdt H,",  
 "usr\_pbgn\_load.tb\_dim\_prmt i,",  
 "usr\_pbgn\_load.tb\_dim\_flal k,",  
 "usr\_pbgn\_load.tb\_dim\_crtr j",  
 " Where r.id\_oprc\_rlzd = t.id\_oprc\_rlzd and",  
 " t.id\_crtr = j.id\_crtr and",  
 " t.id\_flal = k.id\_flal and",  
 " t.id\_cnal\_vnda = a.id\_cnal\_vnda and",  
 " t.id\_oprd = g.id\_oprd and",  
 " t.id\_crdn = b.id\_crdn and",  
 " t.id\_grnt\_rgnl = c.id\_grnt\_rgnl and",  
 " t.id\_grnt = d.id\_grnt and",  
 " t.id\_epdr = e.id\_epdr and",  
 " t.id\_grpo\_prmt = f.id\_grpo\_prmt and",  
 " t.id\_prdt = h.id\_prdt and",  
 " t.id\_prmt = i.id\_prmt and",  
 " r.id\_hist\_fncr = x.id\_hist\_fncr and",  
 " r.dt\_crga >= 20130131 and",  
 " x.cd\_hist\_fncr in ('104','219','220','251','840','218','250')",  
 " Group By round(r.dt\_crga/100),",  
 "r.cd\_tipo\_mvmt,",  
 "x.cd\_hist\_fncr,",  
 "x.ds\_hist\_fncr,",  
 "a.cd\_cnal\_vnda,",  
 "a.ds\_cnal\_vnda,",  
 "g.ds\_oprd,",  
 "b.ds\_crdn,",  
 "d.ds\_grnt,",  
 "c.ds\_grnt\_rgnl,",  
 "e.ds\_epdr,",  
 "f.ds\_grpo\_prmt,",  
 "h.ds\_prdt,",  
 "i.ds\_prmt,",  
 "k.DS\_FLAL,",  
 "j.ds\_crtr,",  
 "j.cd\_crtr,",  
 "round(t.dt\_base/100),",  
 "(round(r.dt\_crga/10000)-round(t.dt\_base/10000))\*12+mod(round(r.dt\_crga/100),100)-mod(round(t.dt\_base/100),100);"  
  
 )  
  
 # executa a consulta  
 # a consulta abaixo roda em média em 17 minutos no Windows Cetelem 32b  
 df\_temp\_liquidacao <- sqlQuery(channel,cSQL\_liq, errors = TRUE)  
  
 # fecha conexao com Oracle  
 odbcClose(channel)  
  
 # força para todos os registros:  
 # condição de ATRASO = "Em Dia"  
 # FAIXA REPORT = "R0"  
 # acumula ocorrências de código de consignado fora dos grupos como geral 99-CONSIGNADO  
 df\_temp\_liquidacao <-  
 df\_temp\_liquidacao %>%  
 mutate (ATRASO = "Em Dia",  
 FAIXA\_REPORT = "R0",  
 id\_faix\_atrs = -1,  
 CD\_CRTR = ifelse(!(CD\_CRTR\_TEMP %in% c(50,52,53,55)), 99, CD\_CRTR\_TEMP))  
  
 # ------------ PASSO ADICIONAL PARA TEMP\_LIQUIDACAO  
  
 # muda vírgula para ponto decimal somente nos campos decimais  
 df\_temp\_liquidacao <- data.frame(lapply(df\_temp\_liquidacao, function(x) gsub(",", ".", x, fixed = TRUE)), stringsAsFactors = FALSE)  
  
 # entrada df\_temp\_liquidacao  
 # saida: df\_riscbgn\_cubo\_liquidacao  
 # sumariza os valores de liquidação, taxa de liquidação e prazo de liquidação de produção por  
 # data de referência   
 # data de processamento  
 # canal de venda  
 # empresa cliente (?)  
 # filial BGN  
 # correspondente bancário  
 # produto  
 # tipo de consignado (?)  
 # faixa de atraso  
 # faixa de report  
 # identificação da faixa de atraso  
 # MOB (?)  
 df\_riscbgn\_cubo\_liquidacao <-  
 df\_temp\_liquidacao %>%  
 select (DT\_REF,DT\_BASE,DS\_CNAL\_VNDA,DS\_EPDR,DS\_FLAL,DS\_GRPO\_PRMT,  
 DS\_PRDT,DS\_CRTR,ATRASO,FAIXA\_REPORT,id\_faix\_atrs, MOB,  
 PRODUCAO, VL\_MVMT, VL\_MVMT\_TX) %>%  
 group\_by(DT\_REF,DT\_BASE,DS\_CNAL\_VNDA,DS\_EPDR,DS\_FLAL,DS\_GRPO\_PRMT,  
 DS\_PRDT,DS\_CRTR,ATRASO,FAIXA\_REPORT,id\_faix\_atrs, MOB) %>%  
 summarize(VLR\_PROD\_LIQ = sum(as.numeric(PRODUCAO)),  
 VLR\_LIQ = sum(as.numeric(VL\_MVMT)),  
 VLR\_LIQ\_TAX = sum(as.numeric(VL\_MVMT\_TX)),  
 VLR\_PROD\_LIQ\_PRAZO = sum(as.numeric(PRODUCAO)\*as.numeric(MOB))) %>%  
 rename(Prazo\_med\_liq = MOB)  
  
 # aqui gravar arquivo raw em riscbgn\_cubo\_liquidacao\_AAAAMMDD\_raw.csv na pasta do mês de processamento  
 # este arquivo deve substituir os dados da aba base usada na planilha TDB Produção  
 write.csv2(df\_riscbgn\_cubo\_liquidacao, file = fileout\_cuboliqu)  
  
 # remove bases usadas  
 rm(df\_temp\_liquidacao, df\_riscbgn\_cubo\_liquidacao)

# ----------------------------------------------  
 # TABELA CARTEIRA  
 # ----------------------------------------------  
  
 # conecta ao Oracle  
 channel <- odbcConnect(caminho,uid=userid, pwd=passwd, believeNRows=FALSE)  
   
 # cria tabela riscbgn.base\_carteira a partir do Oracle, segundo pesquisa sas já existente  
 # em consignado.sas  
 # Descrição:  
 # seleciona a partir das tabelas de fatos de Operações mensais e  
 # Operações Realizadas os valores de montante de saldos contábeis não ???  
 # Também obtém as seguinte dimensões para poder realizar as sumarizações e   
 # filtragens:  
 # tb\_dim\_hist\_fncr (histórico financeiro)  
 # tb\_dim\_grpo\_prmt (correspondente bancário)  
 # tb\_dim\_prdt (produto)  
 # tb\_dim\_epdr (empresa cliente?)  
 # tb\_dim\_flal (filial BGN)  
 # tb\_dim\_prmt (Parceiro?)  
 # tb\_dim\_crtr (tipo de consignado?)  
 # tb\_dim\_cnal\_vnda (canal de venda)  
 # período selecionado:  
 # seleciona para id\_ultm\_dia\_mes igual ao último dia do mês de referência processado  
 cSQL\_cart <- paste0("select tb\_fat\_oprc\_mnsl.CD\_OPRC,",  
 "tb\_fat\_oprc\_mnsl.id\_oprc\_rlzd,",  
 "tb\_dim\_grpo\_prmt.ds\_grpo\_prmt,",  
 "tb\_dim\_prdt.ds\_prdt,",  
 "tb\_dim\_crtr.CD\_CRTR as CD\_CRTR\_TEMP,",  
 "tb\_fat\_oprc\_mnsl.id\_faix\_atrs,",  
 "tb\_dim\_epdr.ds\_epdr,",  
 "tb\_dim\_flal.ds\_flal,",  
 "(tb\_fat\_oprc\_mnsl.mt\_sldo\_dvdr/100) as mt\_sldo\_dvdr,",  
 "(tb\_fat\_oprc\_mnsl.mt\_sldo\_cntb\_cdda/100) as mt\_sldo\_cntb\_cdda,",  
 "(tb\_fat\_oprc\_mnsl.mt\_sldo\_cntb\_nao\_cdda/100) as mt\_sldo\_cntb\_nao\_cdda,",  
 "tb\_dim\_prmt.ds\_prmt,",  
 "tb\_dim\_crtr.ds\_crtr,",  
 "tb\_dim\_cnal\_vnda.ds\_cnal\_vnda,",  
 "tb\_fat\_oprc\_mnsl.nr\_dias\_atrs,",  
 "(tb\_fat\_oprc\_rlzd.vl\_taxa\_cl\_ames/100) as vl\_taxa\_cl\_ames,",  
 "tb\_fat\_oprc\_rlzd.dt\_prmr\_vcto,",  
 "tb\_fat\_oprc\_mnsl.id\_ultm\_dia\_mes,",  
 "tb\_fat\_oprc\_mnsl.dt\_ctrl,",  
 "tb\_fat\_oprc\_rlzd.qt\_prcl,",  
 "(tb\_fat\_oprc\_rlzd.vl\_taxa\_ames/100) as vl\_taxa\_ames,",  
 "tb\_fat\_oprc\_mnsl.dt\_fim\_ctrt,",  
 "tb\_fat\_oprc\_mnsl.dt\_base as data\_base",  
 " from usr\_pbgn\_load.tb\_fat\_oprc\_mnsl,",  
 "usr\_pbgn\_load.tb\_fat\_oprc\_rlzd,",  
 "usr\_pbgn\_load.tb\_dim\_prdt,",  
 "usr\_pbgn\_load.tb\_dim\_flal,",  
 "usr\_pbgn\_load.tb\_dim\_epdr,",  
 "usr\_pbgn\_load.tb\_dim\_grpo\_prmt,",  
 "usr\_pbgn\_load.tb\_dim\_prmt,",  
 "usr\_pbgn\_load.tb\_dim\_crtr,",  
 "usr\_pbgn\_load.tb\_dim\_cnal\_vnda",  
 " where tb\_fat\_oprc\_rlzd.id\_oprc\_rlzd = tb\_fat\_oprc\_mnsl.id\_oprc\_rlzd",  
 " and tb\_dim\_grpo\_prmt.id\_grpo\_prmt = tb\_fat\_oprc\_mnsl.id\_grpo\_prmt",  
 " and tb\_dim\_prdt.id\_prdt = tb\_fat\_oprc\_mnsl.id\_prdt",  
 " and tb\_fat\_oprc\_mnsl.id\_epdr = tb\_dim\_epdr.id\_epdr",  
 " and tb\_dim\_flal.id\_flal = tb\_fat\_oprc\_mnsl.id\_flal",  
 " and tb\_dim\_prmt.id\_prmt = tb\_fat\_oprc\_mnsl.id\_prmt",  
 " and tb\_dim\_crtr.id\_crtr = tb\_fat\_oprc\_mnsl.id\_crtr",  
 " and tb\_dim\_cnal\_vnda.id\_cnal\_vnda = tb\_fat\_oprc\_mnsl.id\_cnal\_vnda",  
 " and tb\_fat\_oprc\_mnsl.id\_ultm\_dia\_mes = ", anoMesDia,  
 " and tb\_fat\_oprc\_mnsl.mt\_sldo\_cntb\_nao\_cdda > 0;")  
  
 # executa a consulta  
 # a consulta abaixo roda em média em 7 minutos no Windows Cetelem 32b  
 df\_temp\_carteira <- sqlQuery(channel,cSQL\_cart, errors = TRUE)  
  
 # fecha conexão Oracle   
 odbcClose(channel)  
  
 # ------------ PASSO ADICIONAL PARA TEMP\_CARTEIRA  
  
 # muda vírgula para ponto decimal somente nos campos decimais  
 df\_temp\_carteira <- data.frame(lapply(df\_temp\_carteira, function(x) gsub(",", ".", x, fixed = TRUE)), stringsAsFactors = FALSE)  
  
 df\_temp\_carteira <-  
 df\_temp\_carteira %>%  
 arrange(ID\_OPRC\_RLZD) %>%  
 mutate (SLD\_CARTEIRA = 0,  
 SLD\_PREJUIZO = 0,  
 MT\_BRUTO\_PRAZO = 0,  
 MT\_BRUTO\_PRAZO\_JUROS = 0,  
 JUROS = 0,  
 SLD\_PREJUIZO = ifelse(ID\_FAIX\_ATRS == "10", as.numeric(MT\_SLDO\_CNTB\_NAO\_CDDA),0),  
 SLD\_CARTEIRA = ifelse(ID\_FAIX\_ATRS != "10", as.numeric(MT\_SLDO\_CNTB\_NAO\_CDDA),0),  
 DT\_REF=trunc(as.numeric(ID\_ULTM\_DIA\_MES)/100),  
 DT\_BASE=trunc(as.numeric(DATA\_BASE)/100),  
 DT\_INICIO\_PGTO=trunc(as.numeric(DT\_PRMR\_VCTO)/100),  
 NR\_PRCL\_PEND = as.numeric(QT\_PRCL) - ((trunc(as.numeric(DT\_REF)/100)-trunc(as.numeric(DT\_INICIO\_PGTO)/100))\*12+((as.numeric(DT\_REF) %% 100)-(as.numeric(DT\_INICIO\_PGTO) %% 100))),  
 NR\_PRCL\_PEND = ifelse(NR\_PRCL\_PEND < 0, 0, NR\_PRCL\_PEND),  
 MT\_BRUTO\_PRAZO=(SLD\_CARTEIRA\*NR\_PRCL\_PEND),  
 MT\_BRUTO\_PRAZO\_JUROS=(SLD\_CARTEIRA\*NR\_PRCL\_PEND\*as.numeric(VL\_TAXA\_AMES)),  
 JUROS = ifelse(is.na(NR\_DIAS\_ATRS) | as.numeric(NR\_DIAS\_ATRS) == 0, SLD\_CARTEIRA\*as.numeric(VL\_TAXA\_AMES),JUROS),  
 ATRASO = ifelse(is.na(NR\_DIAS\_ATRS) | as.numeric(NR\_DIAS\_ATRS) == 0, "EM DIA",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) <= 30, " 1-30",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) <= 59, " 31-59",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) <= 90, " 60-90"," > 90")))),  
 FAIXA\_REPORT = ifelse(as.numeric(NR\_DIAS\_ATRS) <= 6, "R0",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) >= 7 & as.numeric(NR\_DIAS\_ATRS) <= 30, "R1",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) >= 31 & as.numeric(NR\_DIAS\_ATRS) <= 60, "R2",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) >= 61 & as.numeric(NR\_DIAS\_ATRS) <= 90, "R3",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) >= 91 & as.numeric(NR\_DIAS\_ATRS) <= 120, "R4",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) >= 121 & as.numeric(NR\_DIAS\_ATRS) <= 150, "R5",  
 ifelse(as.numeric(NR\_DIAS\_ATRS) >= 151 & as.numeric(NR\_DIAS\_ATRS) <= 180, "R6","R7"))))))),  
 CD\_CRTR = ifelse(!(CD\_CRTR\_TEMP %in% c("50","52","53","55")), "99", CD\_CRTR\_TEMP)) %>% # caso codigo CRTR lido temporario nao for da faixa, fica 99  
 select (-(DATA\_BASE)) %>%  
 arrange(DT\_BASE,QT\_PRCL)  
  
 df\_riscbgn\_base\_carteira <-  
 df\_temp\_carteira %>%  
 select (ID\_OPRC\_RLZD, DT\_REF,DT\_BASE,DS\_CNAL\_VNDA,DS\_EPDR,DS\_FLAL,DS\_GRPO\_PRMT,  
 DS\_PRDT,DS\_CRTR,CD\_CRTR, ATRASO,FAIXA\_REPORT,ID\_FAIX\_ATRS,MT\_BRUTO\_PRAZO,  
 MT\_BRUTO\_PRAZO\_JUROS,SLD\_CARTEIRA,JUROS,SLD\_PREJUIZO) %>%  
 group\_by(ID\_OPRC\_RLZD, DT\_REF,DT\_BASE,DS\_CNAL\_VNDA,DS\_EPDR,DS\_FLAL,DS\_GRPO\_PRMT,  
 DS\_PRDT,DS\_CRTR,CD\_CRTR, ATRASO,FAIXA\_REPORT,ID\_FAIX\_ATRS) %>%  
 summarize(MT\_BRUTO\_PRAZO = sum(MT\_BRUTO\_PRAZO),  
 MT\_BRUTO\_PRAZO\_JUROS = sum(MT\_BRUTO\_PRAZO\_JUROS),  
 SLD\_CARTEIRA = sum(SLD\_CARTEIRA),  
 JUROS = sum(JUROS),  
 SLD\_PREJUIZO = sum(SLD\_PREJUIZO))  
  
 # aqui gravar arquivo raw em riscbgn\_base\_carteira\_AAAAMMDD\_raw.csv na pasta do mês de processamento  
 # este arquivo deve ser incrementado aos dados da aba base usada na planilha TDB Produção  
 write.csv2(df\_riscbgn\_base\_carteira, file = fileout\_basecart)

# remove bases usadas antes de sair  
 rm(df\_temp\_carteira, df\_riscbgn\_base\_carteira, df\_riscbgn\_cubo\_liquidacao, df\_riscbgn\_cubo\_producao)