

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")
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 ADESAO 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_GRP0_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_T",
"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_PBG_N_LOAD.TB_FAT_OPRC_RLZD A11,",
" USR_PBG_N_LOAD.TB_DIM_CNAL_VNDA A12,",
" USR_PBG_N_LOAD.TB_DIM_EPDR A13,",
" USR_PBG_N_LOAD.TB_DIM_FLAL A14,",
" USR_PBG_N_LOAD.TB_DIM_PRDT A15,",
" usr_pbg_n_load.tb_dim_prmt a16,",
" USR_PBG_N_LOAD.TB_DIM_GRNT_RGNL A17,",
" USR_PBG_N_LOAD.TB_DIM_CRTR A18,",
" USR_PBG_N_LOAD.TB_DIM_GRNT A19,",
" USR_PBG_N_LOAD.TB_DIM_CRDN A20,",
" USR_PBG_N_LOAD.TB_DIM_OPRD A21,",
" usr_pbg_n_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(SYSD",
" 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 (ATRASSO = "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)))

# ----- PASSO 3 (OK) Valores conferem com SAS
# cria tabela riscbgn.cubo_producao a partir da tabela temp.producao
# entrada df_temp_producao
# saída: 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,ATRASSO,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,ATRASSO,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 processament
# 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_fnrcr (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: 201406
    "r.cd_tipo_mvmt,",
    "x.cd_hist_fnrcr,",
    "x.ds_hist_fnrcr,",
    "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 -> 2014
    "(round(r.dt_crga/10000) - round(t.dt_base/10000)) * 12 + mod(round(r.dt_crga/100),1
    "sum(round(t.MT_OPRC/100,2)) as Producao,",
    "DECODE (x.cd_hist_fnrcr, '104',sum(round(r.vl_mvmt/-100,2)),sum(round((r.vl_mvmt*t.q
    "DECODE (x.cd_hist_fnrcr, '104',sum(round((r.vl_mvmt*t.qt_prcl*t.vl_taxa_ames)/-10000
    "DECODE (x.cd_hist_fnrcr, '104',sum(round((r.vl_mvmt*t.vl_taxa_ames)/-10000,2)),sum(r

```

```

" From usr_pbgn_load.tb_fat_mvmt_fnrcr r,",
"usr_pbgn_load.tb_dim_hist_fnrcr 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_fnrcr = x.id_hist_fnrcr and",
" r.dt_crga >= 20130131 and",
" x.cd_hist_fnrcr in ('104','219','220','251','840','218','250')",
" Group By round(r.dt_crga/100)," ,
"r.cd_tipo_mvmt," ,
"x.cd_hist_fnrcr," ,
"x.ds_hist_fnrcr," ,
"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)-mo

```

)

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)))

# entrada df_temp_liquidacao
# saída: 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_GRP_O_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_GRP_O_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_slido_dvdr/100) as mt_slido_dvdr,",
                    "(tb_fat_oprc_mnsl.mt_slido_cntb_cdda/100) as mt_slido_cntb_cdda,",
                    "(tb_fat_oprc_mnsl.mt_slido_cntb_nao_cdda/100) as mt_slido_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_ctr",
                    "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_slido_cntb_ao_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)))

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_SLIDO_CNTB_NAO_CDDA),0),
          SLD_CARTEIRA = ifelse(ID_FAIX_ATRS != "10", as.numeric(MT_SLIDO_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_PMRM_VCTO)/100),
          NR_PRCL_PEND = as.numeric(QT_PRCL) - ((trunc(as.numeric(DT_REF)/100)-trunc(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),
          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, "R",
        ifelse(as.numeric(NR_DIAS_ATRS) >= 61 & as.numeric(NR_DIAS_ATRS) <= 90, "R",
        ifelse(as.numeric(NR_DIAS_ATRS) >= 91 & as.numeric(NR_DIAS_ATRS) <= 120, "I",
        ifelse(as.numeric(NR_DIAS_ATRS) >= 121 & as.numeric(NR_DIAS_ATRS) <= 150, "I",
        ifelse(as.numeric(NR_DIAS_ATRS) >= 151 & as.numeric(NR_DIAS_ATRS) <= 180, "I",
        CD_CRTR = ifelse(!(CD_CRTR_TEMP %in% c("50", "52", "53", "55")), "99", CD_CRTR_TEMP)) %>%
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 processament
# este arquivo deve ser incrementado aos dados da aba base usada na planilha TDB Produção
write.csv2(df_riscbgn_base_carteira, file = fileout_basecart)

```