#COUNT para contar quantas variáveis são nulas nas tabelas DEFAULT e loans\_detail.

SELECT

COUNT(\*)

FROM

`riscorelativo3.Projeto03.default`

WHERE

user\_id IS NULL

OR default\_flag IS NULL

SELECT

COUNT(\*)

FROM

`riscorelativo3.Projeto03.loans\_detail`

WHERE

user\_id IS NULL

OR more\_90\_days\_overdue IS NULL

OR using\_lines\_not\_secured\_personal\_assets IS NULL

OR number\_times\_delayed\_payment\_loan\_30\_59\_days IS NULL

OR debt\_ratio IS NULL

OR number\_times\_delayed\_payment\_loan\_60\_89\_days IS NULL

#Query para retornar AS variáveis nulas nas tabelas loans\_outstanding e user\_info.

SELECT

\*

FROM

`riscorelativo3.Projeto03.loans\_outstanding`

WHERE

loan\_id IS NULL

OR user\_id IS NULL

OR loan\_type IS NULL

SELECT

\*

FROM

`riscorelativo3.Projeto03.user\_info`

WHERE

user\_id IS NULL

OR age IS NULL

OR sex IS NULL

OR last\_month\_salary IS NULL

OR number\_dependents IS NULL

##Limpeza das tabelas##

#Substitui os nomes das colunas da tabela DEFAULT, criar uma nova coluna com adimplentes e inadimplentes

CREATE OR REPLACE TABLE

`riscorelativo3.Projeto03.default` AS

WITH

defaultt AS (

SELECT

user\_id AS id\_usuario,

CASE

WHEN default\_flag >= 1 THEN 'INADIMPLENTE'

ELSE 'ADIMPLENTE'

END

AS adimplentes\_inadimplentes,

default\_flag AS classificacao\_de\_inadimplencia

FROM

`riscorelativo3.Projeto03.default` )

SELECT

\*

FROM

defaultt;

#Substitui os nomes das colunas da tabela detail, alterar AS colunas float para INT64

CREATE OR REPLACE TABLE

`riscorelativo3.Projeto03.loans\_detail` AS

WITH

detail AS (

SELECT

user\_id AS id\_usuario,

more\_90\_days\_overdue AS atraso\_superior\_90\_dias,

CAST(using\_lines\_not\_secured\_personal\_assets AS INT64) AS linhas\_nao\_protegidas\_por\_bens\_pessoais,

number\_times\_delayed\_payment\_loan\_30\_59\_days AS atraso\_30\_59\_dias,

number\_times\_delayed\_payment\_loan\_60\_89\_days AS atraso\_60\_89\_dias,

CAST(debt\_ratio AS INT64) AS taxa\_de\_endividamento

FROM

`riscorelativo3.Projeto03.loans\_detail` )

SELECT

\*

FROM

detail;

#Substitui os nomes das colunas da tabela outstanding, alterar os dados da coluna loan\_type para os tipos de emprestimos (imovel ou outros).

CREATE OR REPLACE TABLE

`riscorelativo3.Projeto03.loans\_outstanding` AS

WITH

outstanding AS (

SELECT

user\_id AS id\_usuario,

loan\_id AS id\_emprestimo,

CASE

WHEN(loan\_type) = 'real estate' THEN 'IMOVEL'

WHEN(loan\_type) = 'other' THEN 'OUTRO'

WHEN(loan\_type) = 'others' THEN 'OUTRO'

WHEN(loan\_type) = 'REAL ESTATE' THEN 'IMOVEL'

WHEN(loan\_type) = 'Real Estate' THEN 'IMOVEL'

WHEN(loan\_type) = 'Other' THEN 'OUTRO'

WHEN(loan\_type) = 'OTHER' THEN 'OUTRO'

ELSE 'SEM\_EMPRESTIMO'

END

AS tipo\_de\_emprestimo

FROM

`riscorelativo3.Projeto03.loans\_outstanding` )

SELECT

\*

FROM

outstanding;

# Consulta para verificar quantos valores seriam 0 no cálculo da nova variável.

WITH Qtd\_Emprestimo AS (

SELECT

id\_usuario,

COUNT(DISTINCT id\_emprestimo) AS qtd\_emprestimos

FROM `riscorelativo3.Projeto03.loans\_outstanding`

GROUP BY id\_usuario

)

SELECT

id\_usuario,

qtd\_emprestimos

FROM Qtd\_Emprestimo

WHERE qtd\_emprestimos = 0;

# Substitui os nomes das colunas da tabela user\_info, alterar os dados da coluna last\_month\_salary usei o COALESCE para substituir pela mediana de salário e na coluna number\_dependent por 0 onde for nulo. Na primeira subquery eu criei uma nova variavel chamada total\_de\_emprestimos\_por\_id com base na tabela loans\_outstanding e usei o COALESCE para substituir os valores nulos pelo cálculo da mediana criando uma nova coluna chamada qtd\_emprestimos\_por\_usuario e na segunda subquery alterei o tipo de dado da coluna ultimo\_salario para INT64. e um LEFT JOIN ao final para unir as duas tabelas.

CREATE OR REPLACE TABLE

`riscorelativo3.Projeto03.user\_info` AS

WITH

user AS (

SELECT

user\_id AS id\_usuario,

age AS idade,

UPPER(sex) AS genero,

COALESCE(last\_month\_salary, (

SELECT

APPROX\_QUANTILES(last\_month\_salary, 100)[

OFFSET

(50)]

FROM

`riscorelativo3.Projeto03.user\_info`

WHERE

last\_month\_salary IS NOT NULL)) AS ultimo\_salario,

COALESCE(number\_dependents, 0) AS qtd\_dependentes

FROM

`riscorelativo3.Projeto03.user\_info`

),

Total\_emprestimos AS (

SELECT

id\_usuario,

COUNT(DISTINCT id\_emprestimo) AS total\_de\_emprestimos\_por\_id

FROM `riscorelativo3.Projeto03.loans\_outstanding` AS outs

GROUP BY id\_usuario

)

SELECT

user.id\_usuario,

user.idade,

user.genero,

CAST(user.ultimo\_salario AS INT64) AS ultimo\_salario,

user.qtd\_dependentes,

COALESCE(te.total\_de\_emprestimos\_por\_id, (

SELECT

APPROX\_QUANTILES(te.total\_de\_emprestimos\_por\_id, 100)[

OFFSET

(50)]

FROM

Total\_emprestimos AS te

WHERE

te.total\_de\_emprestimos\_por\_id IS NOT NULL)) AS qtd\_emprestimos\_por\_usuario

FROM

user

LEFT JOIN Total\_emprestimos AS te

ON user.id\_usuario = te.id\_usuario;

# Consultas para confirmar se minha subquery está retornando algum valor nulo

SELECT

id\_usuario,

qtd\_emprestimos\_por\_usuario

FROM `riscorelativo3.Projeto03.user\_info`

WHERE qtd\_emprestimos\_por\_usuario IS NULL;

# LEFT JOIN para unir as tabelas limpas e criar uma nova tabela (risco\_relativo)

CREATE TABLE `riscorelativo3.Projeto03.risco\_relativo` AS

WITH risco\_relativo AS (

SELECT

user.id\_usuario,

user.idade,

user.genero,

user.ultimo\_salario,

user.qtd\_emprestimos\_por\_usuario,

qtd\_dependentes,

dft.adimplentes\_inadimplentes,

dft.classificacao\_de\_inadimplencia,

dtl.atraso\_superior\_90\_dias,

dtl.linhas\_nao\_protegidas\_por\_bens\_pessoais,

dtl.atraso\_30\_59\_dias,

dtl.atraso\_60\_89\_dias,

dtl.taxa\_de\_endividamento,

FROM

`riscorelativo3.Projeto03.user\_info` AS user

LEFT JOIN `riscorelativo3.Projeto03.default` AS dft

ON user.id\_usuario = dft.id\_usuario

LEFT JOIN `riscorelativo3.Projeto03.loans\_detail` AS dtl

ON user.id\_usuario = dtl.id\_usuario

)

SELECT \* FROM risco\_relativo;

# Consultas para confirmar se minha nova planilha está retornando algum valor nulo

SELECT \*

FROM `riscorelativo3.Projeto03.risco\_relativo`

WHERE id\_usuario IS NULL

OR idade IS NULL

OR ultimo\_salario IS NULL

OR qtd\_emprestimos\_por\_usuario IS NULL

OR qtd\_dependentes IS NULL

OR classificacao\_de\_inadimplencia IS NULL

OR atraso\_superior\_90\_dias IS NULL

OR linhas\_nao\_protegidas\_por\_bens\_pessoais IS NULL

OR atraso\_30\_59\_dias IS NULL

OR atraso\_60\_89\_dias IS NULL

OR taxa\_de\_endividamento IS NULL;

##verificar Outliers##

# Max Min, Avg, Med e Desvio Padrao.

WITH Total\_emprestimos AS (

SELECT

id\_usuario,

COUNT(DISTINCT id\_emprestimo) AS total\_de\_emprestimos\_por\_id

FROM `riscorelativo3.Projeto03.loans\_outstanding`

GROUP BY id\_usuario

)

SELECT

MIN(total\_de\_emprestimos\_por\_id) AS min\_total\_por\_id,

MAX(total\_de\_emprestimos\_por\_id) AS max\_total\_por\_id,

AVG(total\_de\_emprestimos\_por\_id) AS media\_total\_por\_id,

APPROX\_QUANTILES(total\_de\_emprestimos\_por\_id, 100)[SAFE\_ORDINAL(50)] AS mediana\_total\_por\_id,

STDDEV(total\_de\_emprestimos\_por\_id) AS desvio\_total\_por\_id,

FROM Total\_emprestimos;

#

SELECT

MIN(idade) AS min\_idade,

MAX(idade) AS max\_idade,

AVG(idade) AS media\_idade,

APPROX\_QUANTILES(idade, 100)[SAFE\_ORDINAL(50)] AS mediana\_idade,

STDDEV(idade) AS desvio\_idade,

MIN(ultimo\_salario) AS min\_salario,

MAX(ultimo\_salario) AS max\_salario,

AVG(ultimo\_salario) AS media\_salario,

APPROX\_QUANTILES(ultimo\_salario, 100)[SAFE\_ORDINAL(50)] AS mediana\_salario,

STDDEV(ultimo\_salario) AS desvio\_salario,

MIN(qtd\_dependentes) AS min\_dependentes,

MAX(qtd\_dependentes) AS max\_dependentes,

AVG(qtd\_dependentes) AS media\_dependentes,

APPROX\_QUANTILES(qtd\_dependentes, 100)[SAFE\_ORDINAL(50)] AS mediana\_dependentes,

STDDEV(qtd\_dependentes) AS desvio\_dependentes,

MIN(qtd\_emprestimos\_por\_usuario) AS min\_emprestimo,

MAX(qtd\_emprestimos\_por\_usuario) AS max\_emprestimo,

AVG(qtd\_emprestimos\_por\_usuario) AS media\_emprestimo,

APPROX\_QUANTILES(qtd\_emprestimos\_por\_usuario, 100)[SAFE\_ORDINAL(50)] AS mediana\_emprestimo,

STDDEV(qtd\_emprestimos\_por\_usuario) AS desvio\_emprestimo,

MIN(atraso\_superior\_90\_dias) AS min\_atraso\_90,

MAX(atraso\_superior\_90\_dias) AS max\_atraso\_90,

AVG(atraso\_superior\_90\_dias) AS media\_atraso\_90,

APPROX\_QUANTILES(atraso\_superior\_90\_dias, 100)[SAFE\_ORDINAL(50)] AS mediana\_atraso\_90,

STDDEV(atraso\_superior\_90\_dias) AS desvio\_atraso\_90,

MIN(atraso\_30\_59\_dias) AS min\_atraso\_30\_59,

MAX(atraso\_30\_59\_dias) AS max\_atraso\_30\_59,

AVG(atraso\_30\_59\_dias) AS media\_atraso\_30\_59,

APPROX\_QUANTILES(atraso\_30\_59\_dias, 100)[SAFE\_ORDINAL(50)] AS mediana\_atraso\_30\_59,

STDDEV(atraso\_30\_59\_dias) AS desvio\_atraso\_30\_59,

MIN(atraso\_60\_89\_dias) AS min\_atraso\_60\_89,

MAX(atraso\_60\_89\_dias) AS max\_atraso\_60\_89,

AVG(atraso\_60\_89\_dias) AS media\_atraso\_60\_89,

APPROX\_QUANTILES(atraso\_60\_89\_dias, 100)[SAFE\_ORDINAL(50)] AS mediana\_atraso\_60\_89,

STDDEV(atraso\_60\_89\_dias) AS desvio\_atraso\_60\_89,

MIN(taxa\_de\_endividamento) AS min\_tx\_endividamento,

MAX(taxa\_de\_endividamento) AS max\_tx\_endividamento,

AVG(taxa\_de\_endividamento) AS media\_tx\_endividamento,

APPROX\_QUANTILES(taxa\_de\_endividamento, 100)[SAFE\_ORDINAL(50)] AS mediana\_tx\_endividamento,

STDDEV(taxa\_de\_endividamento) AS desvio\_tx\_endividamento

FROM `riscorelativo3.Projeto03.risco\_relativo`

# Identifiquei o numero 98 nas tres colunas de atraso de pagamento em um total de 63 usuarios, sendo aproximadamente 0,175% do total da nossa base de 36.000 usuarios.

CREATE TABLE `riscorelativo3.Projeto03.risco\_relativo\_98` AS (

SELECT

id\_usuario,

idade,

genero,

ultimo\_salario,

qtd\_dependentes,

qtd\_emprestimos\_por\_usuario,

adimplentes\_inadimplentes,

classificacao\_de\_inadimplencia,

linhas\_nao\_protegidas\_por\_bens\_pessoais,

taxa\_de\_endividamento,

atraso\_30\_59\_dias,

atraso\_60\_89\_dias,

atraso\_superior\_90\_dias

FROM `riscorelativo3.Projeto03.risco\_relativo`

WHERE atraso\_superior\_90\_dias > 10

AND atraso\_30\_59\_dias > 10

AND atraso\_60\_89\_dias > 10

)

# atraso superior a 90 = valor maximo e de 15 vezes

# atraso entre 30 e 59 dias = valor maximo e de 11 vezes

# atraso entre 60 e 89 dias = valor maximo e de 7 vezes

CREATE TABLE `riscorelativo3.Projeto03.risco\_relativo\_` AS (

SELECT

id\_usuario,

idade,

genero,

ultimo\_salario,

qtd\_dependentes,

qtd\_emprestimos\_por\_usuario,

adimplentes\_inadimplentes,

classificacao\_de\_inadimplencia,

linhas\_nao\_protegidas\_por\_bens\_pessoais,

taxa\_de\_endividamento,

atraso\_30\_59\_dias,

atraso\_60\_89\_dias,

atraso\_superior\_90\_dias

FROM `riscorelativo3.Projeto03.risco\_relativo`

WHERE atraso\_superior\_90\_dias <= 70

AND atraso\_30\_59\_dias <= 70

AND atraso\_60\_89\_dias <= 70

)

# Desvio padrão da tabela risco\_relativo\_ e o mais recomendado para uso em caso do banco, ou seja desconsiderei os valores 98 contidos nas colunas de atrasos.

SELECT

MIN(atraso\_superior\_90\_dias) AS min\_atraso\_90,

MAX(atraso\_superior\_90\_dias) AS max\_atraso\_90,

AVG(atraso\_superior\_90\_dias) AS media\_atraso\_90,

APPROX\_QUANTILES(atraso\_superior\_90\_dias, 100)[SAFE\_ORDINAL(50)] AS mediana\_atraso\_90,

STDDEV(atraso\_superior\_90\_dias) AS desvio\_atraso\_90,

MIN(atraso\_30\_59\_dias) AS min\_atraso\_30\_59,

MAX(atraso\_30\_59\_dias) AS max\_atraso\_30\_59,

AVG(atraso\_30\_59\_dias) AS media\_atraso\_30\_59,

APPROX\_QUANTILES(atraso\_30\_59\_dias, 100)[SAFE\_ORDINAL(50)] AS mediana\_atraso\_30\_59,

STDDEV(atraso\_30\_59\_dias) AS desvio\_atraso\_30\_59,

MIN(atraso\_60\_89\_dias) AS min\_atraso\_60\_89,

MAX(atraso\_60\_89\_dias) AS max\_atraso\_60\_89,

AVG(atraso\_60\_89\_dias) AS media\_atraso\_60\_89,

APPROX\_QUANTILES(atraso\_60\_89\_dias, 100)[SAFE\_ORDINAL(50)] AS mediana\_atraso\_60\_89,

STDDEV(atraso\_60\_89\_dias) AS desvio\_atraso\_60\_89,

FROM `riscorelativo3.Projeto03.risco\_relativo\_`

# CORRE para verificar a correlação

SELECT

CORR(idade, classificacao\_de\_inadimplencia) AS hipotese1,

CORR(qtd\_emprestimos\_por\_usuario, classificacao\_de\_inadimplencia) AS hipotese2,

CORR(atraso\_superior\_90\_dias, classificacao\_de\_inadimplencia) AS hipotese3,

CORR(atraso\_superior\_90\_dias, atraso\_30\_59\_dias) AS exemplo,

CORR(qtd\_emprestimos\_por\_usuario, idade) AS emprestimo\_idade,

CORR(qtd\_emprestimos\_por\_usuario, ultimo\_salario) AS emprestimo\_salario,

CORR(qtd\_emprestimos\_por\_usuario, qtd\_dependentes) AS emprestimo\_dependentes

FROM `riscorelativo3.Projeto03.risco\_relativo\_`

# Quartil e segmentação

CREATE OR REPLACE TABLE `riscorelativo3.Projeto03.risco\_relativo\_quartil` AS

WITH quartil AS (

SELECT

id\_usuario, idade, ultimo\_salario, qtd\_emprestimos\_por\_usuario, qtd\_dependentes, classificacao\_de\_inadimplencia, linhas\_nao\_protegidas\_por\_bens\_pessoais, taxa\_de\_endividamento, atraso\_30\_59\_dias, atraso\_60\_89\_dias, atraso\_superior\_90\_dias,

NTILE(4) OVER (ORDER BY idade) AS quartil\_idade,

NTILE(4) OVER (ORDER BY ultimo\_salario) AS quartil\_salario,

NTILE(4) OVER (ORDER BY qtd\_emprestimos\_por\_usuario) AS quartil\_emprestimos,

NTILE(4) OVER (ORDER BY qtd\_dependentes) AS quartil\_dependentes,

NTILE(4) OVER (ORDER BY classificacao\_de\_inadimplencia) AS quartil\_inadimplencia,

NTILE(4) OVER (ORDER BY linhas\_nao\_protegidas\_por\_bens\_pessoais) AS quartil\_bens\_pessoais,

NTILE(4) OVER (ORDER BY taxa\_de\_endividamento) AS quartil\_endividamento,

NTILE(4) OVER (ORDER BY atraso\_30\_59\_dias) AS quartil\_atraso\_30\_59,

NTILE(4) OVER (ORDER BY atraso\_60\_89\_dias) AS quartil\_atraso\_60\_89,

NTILE(4) OVER (ORDER BY atraso\_superior\_90\_dias) AS quartil\_atraso\_superior\_90

FROM `riscorelativo3.Projeto03.risco\_relativo\_`

),

total\_classificacao AS (

SELECT

SUM(classificacao\_de\_inadimplencia) AS sum\_class,

COUNT(\*) AS total

FROM quartil

),

idade\_risco\_relativo AS (

SELECT

quartil\_idade AS quartil,

MIN(idade) AS min\_idade,

MAX(idade) AS max\_idade,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS riscorelativo\_idade

FROM quartil, total\_classificacao

GROUP BY quartil\_idade, total\_classificacao.sum\_class, total\_classificacao.total

),

ultimo\_salario\_risco\_relativo AS (

SELECT

quartil\_salario AS quartil,

MIN(ultimo\_salario) AS min\_salario,

MAX(ultimo\_salario) AS max\_salario,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS riscorelativo\_salario

FROM quartil, total\_classificacao

GROUP BY quartil\_salario, total\_classificacao.sum\_class, total\_classificacao.total

),

qtd\_dependentes\_risco\_relativo AS (

SELECT

quartil\_dependentes AS quartil,

MIN(qtd\_dependentes) AS min\_dependentes,

MAX(qtd\_dependentes) AS max\_dependentes,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS riscorelativo\_dependentes

FROM quartil, total\_classificacao

GROUP BY quartil\_dependentes, total\_classificacao.sum\_class, total\_classificacao.total

),

qtd\_emprestimos\_risco\_relativo AS (

SELECT

quartil\_emprestimos AS quartil,

MIN(qtd\_emprestimos\_por\_usuario) AS min\_emprestimos,

MAX(qtd\_emprestimos\_por\_usuario) AS max\_emprestimos,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS risco\_relativo\_emprestimo

FROM quartil, total\_classificacao

GROUP BY quartil\_emprestimos, total\_classificacao.sum\_class, total\_classificacao.total

),

atraso\_superior\_90\_risco\_relativo AS (

SELECT

quartil\_atraso\_superior\_90 AS quartil,

MIN(atraso\_superior\_90\_dias) AS min\_superior\_90,

MAX(atraso\_superior\_90\_dias) AS max\_superior\_90,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS risco\_relativo\_90

FROM quartil, total\_classificacao

GROUP BY quartil\_atraso\_superior\_90, total\_classificacao.sum\_class, total\_classificacao.total

),

total\_linhas\_risco\_relativo AS (

SELECT

quartil\_bens\_pessoais AS quartil,

MIN(linhas\_nao\_protegidas\_por\_bens\_pessoais) AS min\_total\_linhas,

MAX(linhas\_nao\_protegidas\_por\_bens\_pessoais) AS max\_total\_linhas,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS risco\_relativo\_total\_linhas

FROM quartil, total\_classificacao

GROUP BY quartil\_bens\_pessoais, total\_classificacao.sum\_class, total\_classificacao.total

),

taxa\_de\_endividamento\_risco\_relativo AS (

SELECT

quartil\_endividamento AS quartil,

MIN(taxa\_de\_endividamento) AS min\_endividamento,

MAX(taxa\_de\_endividamento) AS max\_endividamento,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS risco\_relativo\_taxa\_endividamento

FROM quartil, total\_classificacao

GROUP BY quartil\_endividamento, total\_classificacao.sum\_class, total\_classificacao.total

),

atraso\_30\_59\_dias\_risco\_relativo AS (

SELECT

quartil\_atraso\_30\_59 AS quartil,

MIN(atraso\_30\_59\_dias) AS min\_atraso\_30\_59,

MAX(atraso\_30\_59\_dias) AS max\_atraso\_30\_59,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS risco\_relativo\_30\_59

FROM quartil, total\_classificacao

GROUP BY quartil\_atraso\_30\_59, total\_classificacao.sum\_class, total\_classificacao.total

),

atraso\_60\_89\_dias\_risco\_relativo AS (

SELECT

quartil\_atraso\_60\_89 AS quartil,

MIN(atraso\_60\_89\_dias) AS min\_atraso\_60\_89,

MAX(atraso\_60\_89\_dias) AS max\_atraso\_60\_89,

AVG(classificacao\_de\_inadimplencia) / ((total\_classificacao.sum\_class - SUM(quartil.classificacao\_de\_inadimplencia)) / (total\_classificacao.total - COUNT(\*))) AS risco\_relativo\_60\_89

FROM quartil, total\_classificacao

GROUP BY quartil\_atraso\_60\_89, total\_classificacao.sum\_class, total\_classificacao.total

)

SELECT

a.quartil,

a.riscorelativo\_idade AS idade\_risco\_relativo,

a.min\_idade,

a.max\_idade,

CASE

WHEN a.riscorelativo\_idade < 1 THEN 'Bom pagador'

WHEN a.riscorelativo\_idade > 1 THEN 'Mau pagador'

END AS idade\_segmentacao,

b.riscorelativo\_salario AS ultimo\_salario\_risco\_relativo,

b.min\_salario,

b.max\_salario,

CASE

WHEN b.riscorelativo\_salario < 1 THEN 'Bom pagador'

WHEN b.riscorelativo\_salario > 1 THEN 'Mau pagador'

END AS ultimo\_salario\_segmentacao,

c.riscorelativo\_dependentes AS dependentes\_risco\_relativo,

c.min\_dependentes,

c.max\_dependentes,

CASE

WHEN c.riscorelativo\_dependentes < 1 THEN 'Bom pagador'

WHEN c.riscorelativo\_dependentes > 1 THEN 'Mau pagador'

END AS num\_dependentes\_segmentacao,

d.risco\_relativo\_emprestimo AS total\_emprestimos\_risco\_relativo,

d.min\_emprestimos,

d.max\_emprestimos,

CASE

WHEN d.risco\_relativo\_emprestimo < 1 THEN 'Bom pagador'

WHEN d.risco\_relativo\_emprestimo > 1 THEN 'Mau pagador'

END AS total\_emprestimos\_segmentacao,

e.risco\_relativo\_total\_linhas AS linhas\_nao\_protegidas\_por\_bens\_pessoais\_risco\_relativo,

e.min\_total\_linhas,

e.max\_total\_linhas,

CASE

WHEN e.risco\_relativo\_total\_linhas < 1 THEN 'Bom pagador'

WHEN e.risco\_relativo\_total\_linhas > 1 THEN 'Mau pagador'

END AS total\_linhas\_segmentacao,

f.risco\_relativo\_taxa\_endividamento AS taxa\_de\_endividamento\_risco\_relativo,

f.min\_endividamento,

f.max\_endividamento,

CASE

WHEN f.risco\_relativo\_taxa\_endividamento < 1 THEN 'Bom pagador'

WHEN f.risco\_relativo\_taxa\_endividamento > 1 THEN 'Mau pagador'

END AS taxa\_de\_endividamento\_segmentacao,

g.risco\_relativo\_30\_59 AS atraso\_30\_59\_dias\_risco\_relativo,

g.min\_atraso\_30\_59,

g.max\_atraso\_30\_59,

CASE

WHEN g.risco\_relativo\_30\_59 < 1 THEN 'Bom pagador'

WHEN g.risco\_relativo\_30\_59 > 1 THEN 'Mau pagador'

END AS atraso\_30\_59\_dias\_segmentacao,

h.risco\_relativo\_60\_89 AS atraso\_60\_89\_dias\_risco\_relativo,

h.min\_atraso\_60\_89,

h.max\_atraso\_60\_89,

CASE

WHEN h.risco\_relativo\_60\_89 < 1 THEN 'Bom pagador'

WHEN h.risco\_relativo\_60\_89 > 1 THEN 'Mau pagador'

END AS atraso\_60\_89\_dias\_segmentacao,

i.risco\_relativo\_90 AS atraso\_superior\_90\_dias\_risco\_relativo,

i.min\_superior\_90,

i.max\_superior\_90,

CASE

WHEN i.risco\_relativo\_90 < 1 THEN 'Bom pagador'

WHEN i.risco\_relativo\_90 > 1 THEN 'Mau pagador'

END AS atraso\_superior\_90\_dias\_segmentacao

FROM

idade\_risco\_relativo a

JOIN ultimo\_salario\_risco\_relativo b ON a.quartil = b.quartil

JOIN qtd\_dependentes\_risco\_relativo c ON a.quartil = c.quartil

JOIN qtd\_emprestimos\_risco\_relativo d ON a.quartil = d.quartil

JOIN total\_linhas\_risco\_relativo e ON a.quartil = e.quartil

JOIN taxa\_de\_endividamento\_risco\_relativo f ON a.quartil = f.quartil

JOIN atraso\_30\_59\_dias\_risco\_relativo g ON a.quartil = g.quartil

JOIN atraso\_60\_89\_dias\_risco\_relativo h ON a.quartil = h.quartil

JOIN atraso\_superior\_90\_risco\_relativo i ON a.quartil = i.quartil

ORDER BY a.quartil;