**Exercícios para Avaliação 1 – Tipos de dados e funções escalares**

No servidor

**iprejun.sp.gov.br,44444**

usuário **aspprev**

senha **aspprev**

Base de dados **aspprev2\_jundiaisp\_prd**

Baseado na consulta sobre a tabela PESSOA

SELECT S\_NOM\_PESSOA,DT\_NASC\_PESSOA,DT\_FALEC\_PESSOA,S\_SEXO\_PESSOA,S\_CPF\_PESSOA,S\_PISPASEP\_PESSOA FROM PESSOA

 WHERE *xxxxx*

Selecionar as pessoas:

1. Que nasceram em março de 1985

G: MONTH(DT\_NASC\_PESSOA) = 3 AND YEAR(DT\_NASC\_PESSOA) = 1985

I: MONTH(DT\_NASC\_PESSOA) = 3 AND YEAR(DT\_NASC\_PESSOA) = 1985

M: YEAR(DT\_NASC\_PESSOA) = 1985

R: month(DT\_NASC\_PESSOA) = 3 and YEAR(DT\_NASC\_PESSOA) = 1985

L: YEAR(DT\_NASC\_PESSOA)=1985 AND MONTH(DT\_NASC\_PESSOA)=3

: MONTH(DT\_NASC\_PESSOA) = 3 AND YEAR(DT\_NASC\_PESSOA) = 1985

1. Que tem mais de 75 anos HOJE (incluindo quem faz aniversário hoje)

G: DATEDIFF(year,DT\_NASC\_PESSOA,GETDATE()) > 75 -- AND DT\_FALEC\_PESSOA IS NULL (CASO NÃO QUEIRA CONSIDERAR OS FALECIDOS)

I: DATEDIFF(YEAR, DT\_NASC\_PESSOA, GETDATE()) >= 75
OR (DATEDIFF(YEAR, DT\_NASC\_PESSOA, GETDATE()) = 75
AND MONTH(DT\_NASC\_PESSOA) = MONTH(GETDATE())
AND DAY(DT\_NASC\_PESSOA) = DAY(GETDATE()))

M: DATEDIFF(YEAR,DT\_NASC\_PESSOA,GETDATE()) > 75

R: DT\_NASC\_PESSOA <= '19490410'

L: DATEDIFF(DAY,DT\_NASC\_PESSOA, GETDATE())>75

: DT\_NASC\_PESSOA <= {d '1949-04-10'} AND DT\_FALEC\_PESSOA IS NULL

Obs: SELECT DATEDIFF(YEAR,'1949-05-10','2023-04-10'),dbo.FN\_IdadeDatas('1949-05-10','2023-04-10')

1. Cuja idade termina em '5' (5,15,25,35,...) HOJE (incluindo quem faz aniversário hoje)

G: RIGHT(DATEDIFF(year,DT\_NASC\_PESSOA,GETDATE()),1) = 5

I: (YEAR(GETDATE()) - YEAR(DT\_NASC\_PESSOA)) % 10 = 5

M: RIGHT(YEAR(GETDATE()) - YEAR(DT\_NASC\_PESSOA),1) = '5'

R: ----

L: ----

: dbo.FN\_IdadeDatas(DT\_NASC\_PESSOA,GETDATE()) % 10 = 5

1. Do sexo masculino, considerando que são do sexo masculino as que não tem o sexo declarado

G: S\_SEXO\_PESSOA = 'M' OR S\_SEXO\_PESSOA IS NULL

I: S\_SEXO\_PESSOA = 'M' AND S\_SEXO\_PESSOA IS NOT NULL

M: S\_SEXO\_PESSOA = 'M'

R: (S\_SEXO\_PESSOA = 'M' or S\_SEXO\_PESSOA is null)

L: S\_SEXO\_PESSOA NOT IN ('F')

: COALESCE(S\_SEXO\_PESSOA,'M')='M'

1. Cujo penúltimo dígito do CPF é 3

G: SUBSTRING(S\_CPF\_PESSOA,10,1) = 3

I: SUBSTRING(S\_CPF\_PESSOA, 10, 1) = '3'

M: SUBSTRING(S\_CPF\_PESSOA, LEN(S\_CPF\_PESSOA) - 1, 1) = '3'

R: substring(CAST(S\_CPF\_PESSOA AS VARCHAR),10,1) =3

L: SUBSTRING(CAST(S\_CPF\_PESSOA AS VARCHAR),10,1) =3

: SUBSTRING(S\_CPF\_PESSOA, LEN(S\_CPF\_PESSOA) - 1, 1) = '3'

Obs: S\_CPF\_PESSOA é varchar(11), ou seja, pode ter menos de 11 caracteres

1. Que tem PISPASEP declarado com menos de 11 dígitos

G: Len(S\_PISPASEP\_PESSOA) < 11 OR S\_PISPASEP\_PESSOA IS NULL ---SE NECESSÁRIO BUSCAR TAMBÉM NULOS

I: LEN(S\_PISPASEP\_PESSOA) = 11

M: LEN(S\_PISPASEP\_PESSOA) < 11

R: LEN(S\_PISPASEP\_PESSOA) <11

L: LEN(S\_PISPASEP\_PESSOA) <11

: LEN(S\_PISPASEP\_PESSOA) <11

1. Que morreram com menos de 15 anos

G: DATEDIFF(year,DT\_NASC\_PESSOA,DT\_FALEC\_PESSOA) < 15

I: DT\_FALEC\_PESSOA IS NOT NULL AND DATEDIFF(YEAR, DT\_NASC\_PESSOA, DT\_FALEC\_PESSOA) = 15

M: DATEDIFF(YEAR, DT\_NASC\_PESSOA, DT\_FALEC\_PESSOA) < 15

R: DATEDIFF(YEAR, DT\_FALEC\_PESSOA, GETDATE())<15

L: DATEDIFF(YEAR, DT\_FALEC\_PESSOA, GETDATE())<15

: dbo.FN\_IdadeDatas(DT\_NASC\_PESSOA,DT\_FALEC\_PESSOA)<15

1. Com nome escrito com acentuação

G: S\_NOM\_PESSOA <> S\_NOM\_PESSOA COLLATE SQL\_Latin1\_General\_CP1\_CI\_AI

I: S\_NOM\_PESSOA LIKE '%áàâãäéèêëíìîïóòôõöúùûüçÁÀÂÃÄÉÈÊËÍÌÎÏÓÒÔÕÖÚÙÛÜÇ%'

M: S\_NOM\_PESSOA COLLATE Latin1\_General\_BIN LIKE '%[áéíóúâêôãõàç]%'

R: ----

L: ----

: (S\_NOM\_PESSOA Collate SQL\_Latin1\_General\_CP1253\_CI\_AI) COLLATE Latin1\_General\_100\_CI\_AS <> S\_NOM\_PESSOA

1. Que possuem mais de 5 nomes

G: ----

I: LEN(TRIM(S\_NOM\_PESSOA)) - LEN(REPLACE(S\_NOM\_PESSOA, ' ', '')) + 1 > 5

M: LEN(S\_NOM\_PESSOA) - LEN(REPLACE(S\_NOM\_PESSOA, ' ', '')) >= 4

R: ----

L: ----

: LEN(REPLACE(TRIM(S\_NOM\_PESSOA),' ',' ')) - LEN(REPLACE(S\_NOM\_PESSOA, ' ', '')) >= 5

Obs: 'MARIA DE F M SOUZA'

Baseado na consulta sobre a view vw\_INDICES

SELECT \* FROM vw\_Indices WHERE *xxxxx*

1. Selecionar registros onde o IPCA\_Acu\_Ant \* IPCA\_Indice tem diferença de mais de 0.0000000001 com o IPCA\_Acu (=269 linhas)

G: -----

I: ABS(IPCA\_Acu\_Ant \* IPCA\_Indice - IPCA\_Acu) > 0.0000000001

M: ABS(IPCA\_Acu\_Ant \* IPCA\_Indice - IPCA\_Acu) > 0.0000000001

R: ----

L: ----

: ABS(IPCA\_Acu\_Ant \* IPCA\_Indice - IPCA\_Acu) > 0.0000000001