# Etapa 1: Avaliação de conhecimentos em SQL

*1 - Crie uma estrutura de banco de dados contendo no mínimo 4 tabelas e que*

*possua os seguintes recursos (não necessariamente em todas as tabelas):*

*1. a) Chave primária.*

*2. b) Chave estrangeira.*

*3. c) CONSTRAINT para validar um campo inteiro para ser maior que zero.*

*4. d) Campo DATETIME com valor default para o momento da inserção.*

CREATE SCHEMA db\_service\_desk;

CREATE TABLE db\_service\_desk.tb\_prioridade (

id\_prioridade INT NOT NULL AUTO\_INCREMENT,

codigo INT NULL,

descricao VARCHAR(45) NULL,

PRIMARY KEY (id\_prioridade));

CREATE TABLE db\_service\_desk.tb\_status (

id\_status INT NOT NULL AUTO\_INCREMENT,

codigo VARCHAR(45) NULL,

descricao VARCHAR(45) NULL,

PRIMARY KEY (id\_status));

CREATE TABLE db\_service\_desk.tb\_codigo\_finalizacao (

id\_codigo\_finalizacao INT NOT NULL AUTO\_INCREMENT,

codigo VARCHAR(1) NULL,

descricao VARCHAR(45) NULL,

PRIMARY KEY (id\_codigo\_finalizacao));

CREATE TABLE db\_service\_desk.tb\_chamado (

id\_chamado INT NOT NULL AUTO\_INCREMENT,

nome\_solicitante VARCHAR(45) NULL,

data\_abertura DATETIME NOT NULL DEFAULT CURRENT\_TIMESTAMP,

descricao LONGTEXT NULL,

horas\_estimadas INT NOT NULL,

id\_prioridade INT NULL,

id\_status INT NULL,

id\_codigo\_finalizacao INT NULL,

PRIMARY KEY (id\_chamado),

INDEX FK\_prioridade\_idx (id\_prioridade ASC) VISIBLE,

INDEX FK\_status\_idx (id\_status ASC) VISIBLE,

INDEX FK\_codigo\_finalizacao\_idx (id\_codigo\_finalizacao ASC) VISIBLE,

CONSTRAINT MINIMO\_HORAS\_ESTIMADAS

CHECK(HORAS\_ESTIMADAS > 0),

CONSTRAINT FK\_prioridade

FOREIGN KEY (id\_prioridade)

REFERENCES db\_service\_desk.tb\_prioridade (id\_prioridade)

ON DELETE RESTRICT

ON UPDATE CASCADE,

CONSTRAINT FK\_status

FOREIGN KEY (id\_status)

REFERENCES db\_service\_desk.tb\_status (id\_status)

ON DELETE RESTRICT

ON UPDATE CASCADE,

CONSTRAINT FK\_codigo\_finalizacao

FOREIGN KEY (id\_codigo\_finalizacao)

REFERENCES db\_service\_desk.tb\_codigo\_finalizacao (id\_codigo\_finalizacao)

ON DELETE RESTRICT

ON UPDATE CASCADE);

*2 - Insira registros de exemplos nas tabelas criadas.*

INSERT INTO db\_service\_desk.tb\_prioridade

(codigo, descricao)

VALUES

(1,'Alta'),

(2,'Média'),

(3,'Baixa');

INSERT INTO db\_service\_desk.tb\_status

(codigo, descricao)

VALUES

('Novo','Chamado Aberto'),

('Em andamento', 'Chamado sendo tratado'),

('Finalizado', 'O chamado foi finalizado');

INSERT INTO db\_service\_desk.tb\_codigo\_finalizacao

(codigo, descricao)

VALUES

('C', 'O chamado foi cancelado'),

('P', 'Resolvido parcialmente'),

('T', 'Resolvido totalmente'),

('N','Chamado não resolvido');

INSERT INTO db\_service\_desk.tb\_chamado

(nome\_solicitante, data\_abertura, descricao, horas\_estimadas, id\_prioridade, id\_status, id\_codigo\_finalizacao)

VALUES

('Pedro Silva', '2021-06-01 07:30:00', 'Descrição do problema', 24, 1, 3,2),

('João Silva', '2021-06-01 08:20:00', 'Descrição do problema', 48, 2, 3,2),

('Mateus Silva', '2021-06-01 09:10:00', 'Descrição do problema', 48, 2, 3,3),

('José Silva', '2021-06-05 10:00:00', 'Descrição do problema', 60, 3, 3,3),

('Pedro Silva', '2021-06-05 10:20:00', 'Descrição do problema', 60, 3, 3,1),

('José Silva', '2021-06-05 11:10:00', 'Descrição do problema', 60, 3, 2,null),

('João Silva', '2021-06-10 12:00:00', 'Descrição do problema', 48, 2, 2,null),

('Pedro Silva', '2021-06-10 12:50:00', 'Descrição do problema', 48, 2, 2,null),

('Pedro Silva', '2021-06-10 13:50:00', 'Descrição do problema', 24, 1, 2,null),

('Mateus Silva', '2021-06-15 14:40:00', 'Descrição do problema', 48, 2, 1,null),

('José Silva', '2021-06-15 15:30:00', 'Descrição do problema', 60, 3, 1,null),

('José Silva', '2021-06-15 15:50:00', 'Descrição do problema', 60, 3, 1,null);

1. *Retorne 10 registros da tabela que possui o campo DATETIME ordenado de forma decrescente.*

SELECT \*

FROM tb\_chamado

ORDER BY data\_abertura DESC

LIMIT 10;

1. *Retorne todos os registros de todas as tabelas utilizando JOIN.*

SELECT \*

FROM tb\_chamado c

INNER JOIN tb\_prioridade p ON c.id\_prioridade=p.id\_prioridade

INNER JOIN tb\_status s ON c.id\_status=s.id\_status

LEFT JOIN tb\_codigo\_finalizacao f ON c.id\_codigo\_finalizacao=f.id\_codigo\_finalizacao;

*c) Delete todos os registros da tabela que possui o campo DATETIME*

*que foram inseridos em um determinado intervalo de horas.*

DELETE

FROM tb\_chamado

WHERE TIME(data\_abertura) >= '09:00:00' AND TIME(data\_abertura) < '11:00:00';

# Etapa 2: Avaliação de Lógica

*Faça um algoritmo, podendo utilizar pseudolinguagem, para que ao receber um*

*número qualquer apresente a soma do próprio número com os 100 subsequentes.*

*Não há necessidade de requisitar valores da tela, basta que variáveis sejam*

*atribuídas via código.*

**algoritmo** SomaNumeros;

**variáveis**

NumeroInicial, Soma, Contador: Inteiro;

**início**

NumeroInicial := 1;

Soma := NumeroInicial;

**para** Contador **de** 1 **até** 100 **faça**

**início**

NumeroInicial := NumeroInicial + 1;

Soma := Soma + NumeroInicial;

**fim**;

Imprime(Soma);

**fim**.