Turma: TDS04

SENAI CTM | MARINGÁ - PR

TRABALHO BANCO DE DADOS

Luís Galete Faldão e Gabriel Pereira dos Santos

**1 – Baseado no modelo lógico acima, criar a estrutura de banco de dados respeitando as chaves primárias e estrangeiras constantes no diagrama. Todas as chaves primarias devem ser auto\_increment NÃO É NECESSÁRIO ANEXAR O SCRIPT.**

create database trabalho;

use trabalho;

create table categoria (

codigo\_categoria int(4) not null auto\_increment,

descricao varchar(255) not null,

primary key (codigo\_categoria)

);

create table participante (

codigo\_participante int(4) not null auto\_increment,

nome\_participante varchar(60) not null,

endereco varchar(100) not null,

numero varchar(10) not null,

bairro varchar(50) not null,

cidade varchar(50) not null,

estado varchar(2) not null,

email varchar(50) not null,

primary key (codigo\_participante)

);

create table evento (

codigo\_numero int(4) not null auto\_increment,

nome\_evento varchar(60) not null,

local\_evento varchar(50) not null,

data\_evento date not null,

email\_evento varchar(50) not null,

codigo\_categoria int(4) not null,

primary key (codigo\_numero),

foreign key (codigo\_categoria) references categoria(codigo\_categoria)

);

create table tipo\_inscricao (

codigo\_tipo int(4) not null auto\_increment,

codigo\_evento int(4) not null,

valor\_tipo float(6,2) not null,

primary key (codigo\_tipo),

foreign key (codigo\_evento) references evento(codigo\_numero)

);

create table participante\_inscricao (

cod\_part\_insc int(4) not null auto\_increment,

cod\_participante int(4) not null,

codigo\_tipo int(4) not null,

situacao varchar(2) not null,

data\_inscricao date not null,

primary key (cod\_part\_insc),

foreign key (cod\_participante) references participante(codigo\_participante),

foreign key (codigo\_tipo) references tipo\_inscricao(codigo\_tipo)

);

**2 – Alterar o nome da coluna endereço da tabela Participante para ‘logradouro’**

alter table participante change column endereco logradouro varchar(100);

**3 – Inserir os dados nas tabelas de acordo com os dados abaixo:**

insert into categoria (descricao)

values ("congresso"),("simposio"),("feira"),("semana academica"),("encontro");

insert into evento(nome\_evento, local\_evento, data\_evento, email\_evento, codigo\_categoria)

values ('Semana de TI', 'SENAI', '2017-09-13', 'semanati@evento.br', '4'),

('Simpósio de Eletrônica', 'SENAI', '2017-07-20', 'eletronica@teste.com.br', '2'),

('Semana Acadêmica' , 'Faculdade Albert Einstein' , '2017-05-31' , 'albert@einstein.com.br' , '4'),

('Feira metal mecanica' , 'parque de exposicoes' , '2017-09-01' , 'metal@mecanica.com.br' , '3');

insert into tipo\_inscricao(codigo\_evento, valor\_tipo)

values (1, 80), (1, 98), (2, 30), (2, 40), (3, 150), (3, 180), (4, 90), (4, 140);

**4 – Inserir 10 registros nas tabelas participante e participante\_inscricao. (O campo situação deve contar os valores “ABERTO”, “PAGO” ou “CANCELADO”);**

insert into participante (nome\_participante, logradouro, numero, bairro, cidade, estado, email) values

('Kleberson Pedro', 'Rua da jabuticaba', '12', 'Jardim das rosas', 'Maringa' ,'PR', 'klebersonpedro@gmail.com'),

('Jailson Miguel', 'Rua Governador Robson Miguel', '5', 'Jardim das margaridas', 'Rio de Janeiro' ,'RJ', 'jailsonmiguel@gmail.com'),

('Robson JR', 'Rua do abacaxi', '25', 'Jardim das orquideas', 'Atalaia' ,'PR', 'robsonjr@hotmail.com'),

('Tereza Rosa', 'Rua do limao', '3', 'Jardim das rosas', 'Maringa' ,'PR', 'terezarosa@gmail.com'),

('Maria Terezina', 'Rua do bambu', '6', 'Jardim do girassol', 'Paranavai' ,'PR', 'maria412@hotmail.com'),

('Joao Carlos', 'Rua do bambu', '7', 'Jardim do girassol', 'Paranavai' ,'PR', 'joao42@gmail.com'),

('Carlos Joao', 'Rua da cerejeira', '32', 'Jardim da Jasmini', 'Astorga' ,'PR', 'carlosjoao123@gmail.com'),

('Pedro dos Santos', 'Rua dos morangos', '19', 'Jardim branco', 'Maringa' ,'PR', 'pedrosantos321@hotmail.com'),

('Camargo Emerson', 'Rua do mamao', '42', 'Jardim das margaridas', 'Sao paulo' ,'SP', 'camargoemerson@yahoo.com'),

('Gustavo Kunst', 'Rua da manga', '92', 'Jardim jardinho', 'Gramado' ,'RS', 'gugugaiteiro@gmail.com');

insert into participante\_inscricao(cod\_participante, codigo\_tipo, situacao, data\_inscricao) values

('1', '4' , 'PAGO' , '2017-23-09'),

('2' , '4' , 'ABERTO' , '2017-20-09'),

('2' , '6' , 'CANCELADO' , '2017-10-09'),

('2' , '6' , 'ABERTO' , '2017-04-09'),

('5' , '2' , 'PAGO' , '2017-03-09'),

('6' , '2' , 'PAGO' , '2017-01-09'),

('6' , '5' , 'ABERTO' , '2017-22-10'),

('8' , '5' , 'CANCELADO' , '2017-23-11'),

('8' , '1' , 'ABERTO' , '2017-28-09'),

('10' , '1' , 'CANCELADO' , '2017-25-09');

**5 – Inserir 2 registros na tabela eventos**

insert into evento (nome\_evento, local\_evento, data\_evento, email\_evento, codigo\_categoria)

values ('Briga de galos' , 'senai' , '2021-12-10' , 'contato.rinha@senai.com.br' , '1'),

('BabySenai' , 'senai' , '2021-12-10' , 'babysister@hotmail.com.br' , '1');

**6 – Listar o nome, e-mail, endereço do participante e o nome do evento em que ele realizou inscrição.**

select nome\_participante, email, logradouro, nome\_evento from participante

join participante\_inscricao on participante\_inscricao.cod\_participante = participante.codigo\_participante

join tipo\_inscricao on tipo\_inscricao.codigo\_tipo = participante\_inscricao.codigo\_tipo

join evento on evento.codigo\_numero = tipo\_inscricao.codigo\_evento;

**7 – Listar o nome do evento, data da realização, local e o nome do tipo do evento.**

select nome\_evento, data\_evento, local\_evento, descricao from evento join categoria on categoria.codigo\_categoria = evento.codigo\_categoria;

**8 – Listar todos os tipos de evento que não possuem um evento cadastrado.**

select descricao from categoria where descricao not in (select distinct descricao from categoria right join evento on evento.codigo\_categoria = categoria.codigo\_categoria);

**9 – Listar todos os eventos que são do tipo “ENCONTRO”**

select nome\_evento, descricao from evento join categoria on categoria.codigo\_categoria = evento.codigo\_categoria where descricao = 'encontro';

**10 – Listar todos os participantes do evento “Semana de TI” que estão com a situação PAGO**

select nome\_participante, nome\_evento, situacao from participante

join participante\_inscricao on participante\_inscricao.cod\_participante = participante.codigo\_participante

join tipo\_inscricao on tipo\_inscricao.codigo\_tipo = participante\_inscricao.codigo\_tipo

join evento on evento.codigo\_numero = tipo\_inscricao.codigo\_evento where nome\_evento = 'semana de ti' and situacao = 'PA';

**11 - Listar a quantidade de inscritos por evento cuja situação da inscrição não seja CANCELADA**

select count(cod\_part\_insc), nome\_evento from participante\_inscricao join tipo\_inscricao on tipo\_inscricao.codigo\_tipo = participante\_inscricao.codigo\_tipo join evento on evento.codigo\_numero = tipo\_inscricao.codigo\_evento;

**12 – Listar o valor total (SOMA) de todas as inscrições**

select sum(valor\_tipo) from tipo\_inscricao;

**13 - Listar a quantidade de inscritos por tipo de evento**

select count(nome\_participante), descricao from participante join participante\_inscricao on participante\_inscricao.cod\_participante = participante.codigo\_participante join tipo\_inscricao on tipo\_inscricao.codigo\_tipo = participante\_inscricao.codigo\_tipo join evento on evento.codigo\_numero = tipo\_inscricao.codigo\_evento join categoria on categoria.codigo\_categoria = evento.codigo\_categoria group by descricao;

**14 - Listar o evento que será realizado na maior data e o que será realizado na menor data**

select min(data\_evento), max(data\_evento) from evento;

**15 – Listar o nome do participante e o valor pago na inscrição**

select nome\_participante, valor\_tipo from participante join participante\_inscricao on participante\_inscricao.cod\_participante = participante.codigo\_participante join tipo\_inscricao on tipo\_inscricao.codigo\_tipo = participante\_inscricao.codigo\_tipo;

**16 – Listar o total de inscrições por situação.**

select count(cod\_part\_insc), situacao from participante\_inscricao group by situacao;

**17 – Listar a quantidade de eventos por tipo.**

select descricao, COUNT(nome\_evento) as 'quantidade' from categoria join evento on evento.codigo\_categoria = categoria.codigo\_categoria group by descricao;

**18 – Listar os eventos que não possuem participantes inscritos**

select distinct nome\_evento from evento right join tipo\_inscricao on evento.codigo\_numero = tipo\_inscricao.codigo\_evento;

**19 – Listar os participantes que não estão inscritos em nenhum evento**

select nome\_participante from participante left join participante\_inscricao on participante.codigo\_participante = participante\_inscricao.cod\_participante where participante\_inscricao.cod\_participante is null;

**20 – Listar o nome do participante o e-mail das inscrições pagas**

select nome\_participante, email, situacao from participante join participante\_inscricao on participante\_inscricao.cod\_participante = participante.codigo\_participante where situacao = 'PA';

**21 – Alterar o nome da coluna “SITUACAO” para “STATUS” da tabela** participante\_inscricaoalter table participante\_inscricao change situacao status varchar(2);