**Exercício de Fixação – Functions**

**A** - Deseja-se pesquisar a nota média de um determinado aluno, informando o parâmetro (nome).

Utilize a Function do tipo escalar – Retorna um valor.

DDL:

Crie uma tabela “Alunos” com os atributos (IdAluno,Nome,Nota1,Nota2,Nota3,.......)

Crie a função “Media” para retornar a média das notas de um determinado aluno

DML:

Insira os dados na tabela “Alunos” – (crie pelo menos 3 alunos)

DQL:

Realize a consulta da média de um determinado aluno, informando o nome (Utilizar a function)

**B** - Deseja-se pesquisar os nomes dos alunos, cuja a sua 2° nota cadastrada (Nota2) seja maior do que 6.

Utilize a Function do tipo Inline table – Retorna uma tabela.

DDL:

Utilize a mesma tabela anterior

Crie a função “MostrarNota” para retornar os nomes e a 2º nota dos alunos que forem maiores que 6

DML:

Utilize os mesmos dados da tabela anterior

DQL:

Realize a consulta dos alunos cuja 2º nota cadastrada seja maior que 6 (Utilizar a function)

**C** - Deseja-se pesquisar os nomes dos alunos, a 1° nota cadastrada (Nota1) para esse aluno, o curso que faz e qual período estuda.

Utilize a Function do tipo Multi-statement – Retorna uma tabela.

DDL:

Aproveite a tabela “Alunos” já existente

Crie uma tabela “DadosAlunos” com os atributos (IdDadoAluno,IdAluno,Nome,Curso,Período)

Crie a função “MultiTabelas” para retornar todos os nomes dos alunos, sua 1º nota cadastrada ,qual curso faz e qual período estuda.

DML:

Insira os dados na tabela “DadosAlunos”

DQL:

Realize a consulta dos alunos mostrando sua 1º nota,nome,curso e periodo (Utilizar a function)

**GABARITO**

A –

DDL

CREATE DATABASE Senai;

USE Senai;

CREATE TABLE Alunos

(

IdAluno INT PRIMARY KEY IDENTITY

,Nome VARCHAR(100) NOT NULL

,Nota1 TINYINT NOT NULL

,Nota2 TINYINT NOT NULL

,Nota3 TINYINT NOT NULL

,Nota4 TINYINT NOT NULL

,Nota5 TINYINT NOT NULL

);

CREATE FUNCTION Media (@Nome VARCHAR(50))

RETURNS REAL

AS

BEGIN

DECLARE @Media REAL

SELECT @Media = (Nota1 + Nota2 + Nota3 + Nota4 + Nota5)/5

FROM Alunos

WHERE Alunos.Nome = @Nome

RETURN @Media

END

DML

USE Senai;

INSERT INTO Alunos (Nome,Nota1,Nota2,Nota3,Nota4,Nota5)

VALUES ('Carlos',7,8,9,10,5)

,('Leandro',9,9,9,10,6)

,('Enzo',5,9,10,5,8)

,('João',8,9,10,5,4)

,('Camila',7,8,9,10,8)

,('Rafael',9,9,9,10,8);

DQL

SELECT dbo.Media ('Camila') AS [Média];

------------------------------------------------------------------------------------------------------------------------------

B –

DDL

CREATE FUNCTION MostrarNota (@Valor REAL)

RETURNS TABLE

AS

RETURN

(

SELECT Alunos.Nome,Alunos.Nota2

FROM Alunos

WHERE Alunos.Nota2 > @Valor );

DQL

SELECT Nome,Nota5 FROM MostrarNota (6);

------------------------------------------------------------------------------------------------------------------------------

C –

DDL

CREATE TABLE DadosAlunos

(

IdDadoAluno INT PRIMARY KEY IDENTITY

,IdAluno INT FOREIGN KEY REFERENCES Alunos(IdAluno)

,Curso VARCHAR(100)

,Periodo VARCHAR(100)

);

CREATE FUNCTION MultiTabelas()

RETURNS @Valores TABLE

(Nome VARCHAR(100),Nota1 TINYINT,Curso VARCHAR(100),Periodo VARCHAR(100))

AS

BEGIN

INSERT @Valores(Nome,Nota1,Curso,Periodo)

SELECT

Alunos.Nome,Alunos.Nota1,DadosAlunos.Curso,DadosAlunos.Periodo

FROM Alunos

INNER JOIN DadosAlunos

ON Alunos.IdAluno = DadosAlunos.IdAluno;

RETURN

END

DML

INSERT INTO DadosAlunos (IdAluno,Curso,Periodo)

VALUES (1,'DEV','Manhã')

,(2,'Redes','Manhã')

,(3,'DEV','Tarde')

,(4,'Design','Tarde')

,(5,'Mobile','Manhã')

,(6,'Excel','Manhã');

DQL

SELECT \* FROM MultiTabelas ();

------------------------------------------------------------------------------------------------------------------------------