**Levantamento de Consultas**

O sistema deve informar:

1. Todas as negociações que aluno João de Sousa fez. O resultado deve conter o código e o nome do aluno, o codigo da compra, a descrição, o valor e a data da referente.

SELECT A.Nome, N.CodigoAluno, N.Codigo, N.Descricao, N.Valor, N.Data

FROM Negociacao N, Aluno A

WHERE A.Codigo = N.CodigoAluno AND

A.Nome = 'João de Sousa'

1. Todas as negociações realizadas por cada gerente, ordenando-os por sexo e nome, selecionando o código, nome e sexo, e o código, a descrição, o valor e a data da compra referente.

SELECT GRN.CodigoGerente, G.Nome, G.Sexo, GRN.CodigoNegociacao, N.Descricao, N.Valor, GRN.Data

FROM Gerente\_Realiza\_Negociacao GRN, Negociacao N, Gerente G

Where G.Codigo = GRN.CodigoGerente AND

N.Codigo = GRN.CodigoNegociacao

ORDER BY G.Sexo, G.Nome

1. Selecione o código e nome do aluno, junto ao código, descrição, valor e data da negociação que estão pendentes de pagamento pelo aluno João de Sousa.

SELECT N.CodigoAluno, A.Nome, N.Codigo AS CodigoNegociacao, N.Descricao, N.Valor, N.Data

FROM ALUNO A, Negociacao N JOIN Pagamento P ON N.Codigo = P.CodigoNegociacao AND P.Quitada = false

WHERE A.Codigo = N.CodigoAluno AND

A.Nome = 'João de Sousa'

1. O código, nome e sexo do gerente, junto ao código, descrição, valor e data da negociação que estão pendentes de pagamento pelo gerente de nome Felipe Augusto Vilela.

SELECT G.Codigo AS Gerente, G.Nome, G.Sexo, N.Codigo AS CodigoNegociacao, N.Descricao, N.Valor, GRN.Data

FROM GERENTE G, NEGOCIACAO N, PAGAMENTO P, GERENTE\_Realiza\_NEGOCIACAO GRN

WHERE G.Codigo = GRN.CodigoGerente AND

GRN.CodigoNegociacao = N.Codigo AND

P.CodigoNegociacao = N.Codigo AND

P.CodigoNegociacao = GRN.CodigoNegociacao AND

P.Quitada = false AND

GRN.CodigoNegociacao IN

(SELECT Codigo

FROM GERENTE

WHERE Nome = 'Valesca Maria da Costa')

1. Todos os dados de produto, com o código da negociação, a descrição e o valor, junto ao código, nome, e sexo do aluno Maria do Carmo.

SELECT A.Codigo AS Aluno, A.Nome, A.Sexo, N.Codigo AS Negociacao, N.Descricao, N.Valor, P.Codigo AS Produto, P.Nome, P.Preco, P.Marca, P.Quantidade

FROM Negociacao N, Aluno A, Produto P, Negociacao\_Contem\_Produto NCP

WHERE A.Nome = 'João de Sousa' AND

N.Codigo = NCP.CodigoNegociacao AND

P.Codigo = NCP.CodigoProduto AND

N.CodigoAluno = A.Codigo

1. Todas as informações dos produtos comprados por Felipe Augusto Vilela. A consulta deve retornar o nome, o código e sexo do gerente. Também deve retornar cada negociação, com o código, descrição e valor junto a data da realização da negociação.

SELECT G.Nome AS Gerente, G.Sexo, N.Descricao, N.Valor, GRN.Data, P.Nome, P.Preco, P.Quantidade

FROM GERENTE G, NEGOCIACAO N, GERENTE\_Realiza\_NEGOCIACAO GRN, PRODUTO P, NEGOCIACAO\_Contem\_PRODUTO NCP

WHERE P.Codigo = NCP.CodigoProduto AND

GRN.CodigoNegociacao = NCP.CodigoNegociacao AND

GRN.CodigoNegociacao = N.Codigo AND

GRN.CodigoGerente IN

(SELECT Codigo

FROM GERENTE

WHERE Nome = 'Felipe Augusto Vilela')

1. O código e o nome de todos os alunos, junto ao valor, data de pagamento previsto e a data da realização do pagamento de fato da mensalidade.

SELECT al.Codigo, al.Nome, me.Valor, me.DataPagamento AS DataPagamentoPrevista, me.DataRecibo AS DataRealizacaoPagamento

FROM ALUNO al, MENSALIDADE me

WHERE al.Codigo = me.CodigoAluno

1. Os dados das compras, com o nome de todos os referidos alunos, mesmo que o aluno ainda não tenha feito compra.

SELECT C.Codigo, C.CodigoAluno, C.CNPJouCPF, C.Descriçao, C.Valor, C.Data, A.Nome

FROM (Compra C LEFT OUTER JOIN Aluno A ON C.CodigoAluno = A.Codigo)

1. Informações de medidas, por uma determinada data, do aluno Joana Pereira de Sousa.

SELECT \*

FROM MEDIDAS me

WHERE me.Codigo IN

(SELECT pa.CodigoMedidas

FROM PROFESSOR\_ALUNO pa

WHERE pa.Data = '2017-04-05' AND

pa.CodigoAluno IN

(SELECT Codigo

FROM ALUNO

WHERE Nome = 'Joana Pereira de Sousa'))

1. As características dos exercícios do treino do aluno Pedro da Silva. A consulta deverá conter o nome do aluno procurado, o tipo de treino que está treinando, a data que começou a treinar e o tempo que o treino deve durar, a descrição do treino e as informações dos exercícios de cada dia (de cada siga do tipo).

SELECT al.Nome, tr.Tipo AS TipoTreino, alt.DataInicio, alt.Duraçao, tr.Descriçao, ex.Codigo AS CodigoExercicio, ex.Tipo, ex.Series, ex.Repetiçoes, ex.Nome, ex.Musculo, ex.DropSet

FROM ALUNO al, TREINO tr, ALUNO\_Tem\_TREINO alt, EXERCICIO ex, TREINO\_Tem\_EXERCICIO tre

WHERE alt.CodigoTreino = tr.Codigo AND

tre.CodigoExercicio = ex.Codigo AND

tre.CodigoTreino = alt.CodigoTreino AND

alt.CodigoAluno = al.Codigo AND

al.Codigo IN

(SELECT Codigo

FROM ALUNO

WHERE Nome = 'João de Sousa')

1. O nome e a matricula de todos os alunos que adquiriram em alguma compra o produto ‘Whey Protein’, não importando a marca.

(SELECT A.Nome, M.Codigo AS Matricula

FROM Aluno A, Matricula M, Compra C, Produto P, Compra\_Contem\_Produto CCP

WHERE P.Nome = 'Whey Protein' AND

A.Codigo = C.CodigoAluno AND

P.Codigo = CCP.CodigoProduto AND

C.Codigo = CCP.CodigoCompra AND

A.Codigo = M.CodigoAluno)

INTERSECT

(SELECT A.Nome, M.Codigo AS MAtricula

FROM Aluno A, Matricula M)

1. O salário, o nome e o sexo dos professores.

SELECT Nome, Sexo, Salario

FROM PROFESSOR

ORDER BY Nome, Sexo, Salario

1. A quantidade de alunos ativos na academia no momento da consulta do sistema.

SELECT COUNT(\*) NumeroAlunosAtivos

FROM MATRICULA

WHERE DataTrancamento IS NULL OR

DataTrancamento > (SELECT CURRENT\_DATE)

1. O código e o nome do aluno com todos os nomes dos produtos já comprados por ele.

(SELECT A.Codigo, A.Nome AS NomeAluno, P.Nome

FROM Aluno A, Produto P, Compra C, Compra\_Contem\_Produto CCP

WHERE A.Codigo = C.CodigoAluno AND

P.Codigo = CCP.CodigoProduto AND

C.Codigo = CCP.CodigoCompra

GROUP BY A.Codigo, A.Nome, P.Nome)

INTERSECT

(SELECT A.Codigo, A.Nome, P.Nome

FROM Aluno A, Produto P)

ORDER BY NomeAluno

1. Qual tipo de treino e o professor que montou o treino de João de Sousa. A consulta deve retornar o nome do professor, o nome do aluno, o tipo de treino de João e a descrição do tipo de treino.

SELECT pr.Nome AS Professor, al.Nome AS Aluno, tr.Tipo AS TipoTreino, tr.Descriçao

FROM ALUNO al, TREINO tr, ALUNO\_Tem\_TREINO alt, PROFESSOR pr, PROFESSOR\_EXERCICIO pe

WHERE alt.CodigoTreino = pe.CodigoTreino AND

pe.CodigoProfessor = pr.Codigo AND

alt.CodigoAluno = al.Codigo AND

tr.Codigo = alt.CodigoTreino AND

al.Codigo IN

(SELECT Codigo

FROM ALUNO

WHERE Nome = 'Pedro da Silva')

1. O aluno que ainda não pagou a mensalidade de um determinado mês, sendo estando com atraso ou adiantamento no pagamento. (Com IS NULL).
2. Todos os alunos que tem o primeiro nome que começam com a letra “J’ ou “R”.
3. Os professores que tenham o primeiro nome começando com “Andr”, que sejam do sexo feminino e ganham mais de R$ 1.000,00.
4. Os alunos que fizeram alguma compra e ainda não pagaram totalmente. (Use exists e not exists).
5. Os gerentes que fizeram alguma compra e ainda não pagaram totalmente. (Use exists e not exists).
6. A quantidade de alunos do sexo masculino. (Use count e group by).

PROCEDIMENTOS ARMAZENADOS

1. Função que calcule o BF de todos os alunos.

CREATE OR REPLACE FUNCTION BF(INT)

RETURNS REAL AS '

DECLARE

aluno ALUNO%ROWTYPE;

matricula ALIAS FOR $1;

BEGIN

SELECT INTO aluno \* FROM ALUNO

WHERE Codigo = matricula;

IF(aluno.Sexo = ''M'') THEN

RETURN ((1.20\*IMC(aluno.Codigo))+(0.23\*aluno.Idade)-(10.8\*1)-5.4);

END IF;

RETURN ((1.20\*IMC(aluno.Codigo))+(0.23\*aluno.Idade)-(10.8\*0)-5.4);

END

' LANGUAGE plpgsql;

1. Função que calcule o IMC de todos os alunos.

CREATE OR REPLACE FUNCTION IMC(INT)

RETURNS REAL AS '

DECLARE

aluno ALUNO%ROWTYPE;

matricula ALIAS FOR $1;

BEGIN

SELECT INTO aluno \* FROM ALUNO

WHERE Codigo = matricula;

RETURN aluno.peso/(aluno.altura\*aluno.altura);

END

' LANGUAGE plpgsql;