**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 Valesca Maria da Costa.

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 = GRN.CodigoNegociacao AND

P.Quitada = false AND

GRN.CodigoGerente 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 João de Sousa.

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. 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 A.Codigo, A.Nome, M.Valor, M.DataPagamento AS DataPagamentoPrevista, M.DataRecibo AS DataRealizacaoPagamento

FROM ALUNO A, MENSALIDADE M

WHERE A.Codigo = M.CodigoAluno

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

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

FROM (ALUNO A LEFT OUTER JOIN NEGOCIACAO N ON A.Codigo = N.CodigoAluno)

1. Informações de medidas, por uma determinada data: 2017-04-05, do aluno Joana Pereira de Sousa.

SELECT \*

FROM MEDIDAS M

WHERE M.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 João de Sousa. 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 A.Nome, T.Tipo AS TipoTreino, ATT.DataInicio, ATT.Duracao, T.Descricao, E.Codigo AS CodigoExercicio, E.Tipo, E.Series, E.Repeticoes, E.Nome, E.Musculo, E.DropSet

FROM ALUNO A, TREINO T, ALUNO\_Tem\_TREINO ATT, EXERCICIO E, TREINO\_Tem\_EXERCICIO TTE

WHERE ATT.CodigoTreino = T.Codigo AND

TTE.CodigoExercicio = E.Codigo AND

TTE.CodigoTreino = ATT.CodigoTreino AND

ATT.CodigoAluno = A.Codigo AND

A.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 negociação o produto Whey Protein, não importando a marca.

(SELECT A.Nome, M.Codigo AS Matricula

FROM ALUNO A, MATRICULA M, NEGOCIACAO N, PRODUTO P, NEGOCIACAO\_CONTEM\_PRODUTO NCP

WHERE P.Nome = 'Whey Protein' AND

A.Codigo = N.CodigoAluno AND

P.Codigo = NCP.CodigoProduto AND

N.Codigo = NCP.CodigoNegociacao 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 ordenados pelo sexo e nome.

SELECT Nome, Sexo, Salario

FROM PROFESSOR

ORDER BY Sexo, Nome

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. Os códigos e o nomes dos alunos, dispostos em ordem alfabética ascendente, agrupados por código, nome do aluno e do produto, com todos os nomes dos produtos já negociados por ele.

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

FROM ALUNO A, PRODUTO P, NEGOCIACAO N, Negociacao\_Contem\_Produto NCP

WHERE A.Codigo = N.CodigoAluno AND

P.Codigo = NCP.CodigoProduto AND

N.Codigo = NCP.CodigoNegociacao

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 Pedro da Silva. A consulta deve retornar o nome do professor, o nome do aluno, o tipo de treino de Pedro e a descrição do tipo de treino.

SELECT P.Nome AS Professor, A.Nome AS Aluno, T.Tipo AS TipoTreino, T.Descricao

FROM ALUNO A, TREINO T, ALUNO\_Tem\_TREINO ATT, PROFESSOR P, PROFESSOR\_EXERCICIO PE

WHERE ATT.CodigoTreino = PE.CodigoTreino AND

PE.CodigoProfessor = P.Codigo AND

ATT.CodigoAluno = A.Codigo AND

T.Codigo = ATT.CodigoTreino AND

A.Codigo IN

(SELECT Codigo

FROM ALUNO

WHERE Nome = 'Pedro da Silva')

1. Os alunos que ainda não pagaram a mensalidade de um determinado mês.

SELECT A.Nome

FROM ALUNO A, MENSALIDADE M

WHERE A.Codigo = M.CodigoAluno AND M.DataPagamento IS NULL

1. Nome de todos os alunos que tem o primeiro nome começando com a letra “J” ou “R”.

SELECT A.Nome

FROM ALUNO A

WHERE A.Nome LIKE 'J%' OR A.Nome LIKE 'R%'

1. O nome dos professores que tenham o primeiro nome começando com “Andr”, que sejam do sexo feminino e ganham mais de R$ 1.000,00.

SELECT P.Nome

FROM PROFESSOR P

WHERE P.Nome LIKE 'Andr%' AND P.Sexo = 'F' AND P.Salario > 1000

1. Nome dos alunos que fizeram alguma negociação e ainda não pagaram totalmente.

SELECT A.Nome

FROM ALUNO A

WHERE EXISTS

(SELECT \* FROM NEGOCIACAO N

WHERE A.Codigo = N.CodigoAluno AND N.Tipo = 'Venda' AND NOT EXISTS

(SELECT \* FROM PAGAMENTO P

WHERE N.Codigo = P.CodigoNegociacao AND P.Quitada = true))

1. Os gerentes que fizeram alguma negociação e ainda não pagaram totalmente, eliminando resultados repetidos.

SELECT DISTINCT G.Nome

FROM GERENTE G, GERENTE\_Realiza\_NEGOCIACAO GRN

WHERE EXISTS

(SELECT \* FROM NEGOCIACAO N

WHERE G.Codigo = GRN.CodigoGerente AND N.Codigo = GRN.CodigoNegociacao AND N.Tipo = 'Compra' AND NOT EXISTS

(SELECT \* FROM PAGAMENTO P

WHERE GRN.CodigoNegociacao = P.CodigoNegociacao AND P.Quitada = true))

1. A quantidade de alunos do sexo masculino agrupados por grupo.

SELECT COUNT(\*) AS QuantidadeAlunosSexoMasculino

FROM ALUNO A

WHERE A.Sexo = 'M'

GROUP BY A.Sexo

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;