



**Disciplina: Banco de Dados II**

**Professora: Damires e Thiago**

**Aluno:** \_\_\_\_\_ **Matrícula:** \_\_\_\_\_

## **Tarefa 8 - Funções armazenadas**

**\*\* Use o BD *Pedidos***

1. Transforme o bloco anônimo “*atualiza\_status\_estoque.sql*” (material 10-PL/SQL INTRO), mostrado a seguir, em uma função armazenada que receba o **código do produto** como parâmetro e faça a **atualização do status do produto passado**.

```
Do $$  
Declare qtd_atual produto.quantest%type;  
Begin  
select quantest into qtd_atual from produto  
where codprod = 1;  
if qtd_atual > 30 then  
update produto  
set status = 'Estoque dentro do esperado'  
where codprod = 1;  
else  
update produto  
set status = 'Estoque fora do limite minimo'  
where codprod = 1;  
end if;  
End$$;
```

Crie a função pedida e execute-a seguindo os comandos:

```
select * from produto;  
  
update produto set status = null where codprod = 10;  
  
select atualizaStatus(10);
```

**\*\* Altere o código do produto conforme seus dados.**

2. Analise e execute a função seguinte. Explique o que ela faz.

```
create or replace function getSumSalario()  
returns numeric  
as $$  
Declare  
salcomp numeric;  
v record;  
Begin  
Salcomp = 0;  
for v in (select salariofixo from vendedor where salariofixo is not null) loop
```

```

salcomp = salcomp + v.salariofixo;
end loop;
return salcomp;
end;
$$ LANGUAGE 'plpgsql';

select getSumsalario();

```

2.1 Crie outra versão desta function (getSumSalario2), usando, dessa vez, a função **SUM** SQL pré-definida. Mostre sua execução.

3. Crie uma tabela **Fornecedor** (esquema de metadados mostrado a seguir). Em seguida, faça uma função que permita a inserção de registros nesta tabela, passando os parâmetros necessários (**SEM** o código que é tipo serial).

fornecedor						
General Columns Advanced Constraints Parameters Security SQL						
Inherited from table(s) Select to inherit from...						
Columns +						
	Name	Data type	Length/Precision	Scale	Not NULL?	Primary key?
	cod	integer			<input checked="" type="checkbox"/> Yes	<input checked="" type="checkbox"/> Yes
	nome	character varying	30		<input type="checkbox"/> No	<input type="checkbox"/> No
	cnpj	character varying	15		<input type="checkbox"/> No	<input type="checkbox"/> No
	email	character varying	15		<input type="checkbox"/> No	<input type="checkbox"/> No

\*\* Execute a função e insira 3 fornecedores. Consulte a tabela e veja como ficaram os registros.

4. Desenvolva uma função armazenada (showFornecedor) que mostre, via mensagem, todos os fornecedores cadastrados. Apresente seus códigos, nomes e emails. Use um **cursor** para isso. Mostre sua execução. O resultado deve ser mostrado assim:

Data Output	Explain	Messages	Notifications
NOTICE: Dados do Fornecedor: 1--Fabrica1--xxx@gmail.com			
NOTICE: Dados do Fornecedor: 2--Fabrica2--xxx@gmail.com			
NOTICE: Dados do Fornecedor: 3--Fabrica3--xxx@gmail.com			

\* Dica: use a função de concatenação de string ||