

# CK0114 - Fundamentos de Bancos de Dados - 2023

Italo Abreu, Rafael Sá, Javam Machado

## Trabalho Prático II Desenvolvimento de aplicações para Bancos de Dados

### 1 Atividade

A atividade consiste em introduzir para os alunos o desenvolvimento de aplicações para bancos de dados. Nessa atividade, os alunos irão utilizar Python para desenvolver uma aplicação cliente que consome o serviço de banco de dados (PostgreSQL) a fim de realizar consultas, criar tabelas, inserir/deletar dados e fazer chamadas a procedimentos armazenados.

### 2 Instruções

#### 2.1 Instalando Python 3

Caso ainda não tenha Python 3 instalado na sua máquina, você poderá instalar de acordo com o seguinte tutorial:

1. **Windows:** <https://python.org.br/instalacao-windows/>
2. **Linux:** <https://python.org.br/instalacao-linux/>

Em seguida instale o módulo *psycopg2*:

```
pip install psycopg2
```

Acesse a documentação do *psycopg2* para detalhes e guias de uso.

Os dados para conexão com o bancos de dados são os seguintes:

- host: 200.129.44.249
- database: O mesmo usado para o Trabalho Prático I
- user: *suaMatricula*
- Senha: *suaMatricula@fbd*

**Importante:** Para interação com o banco de dados via Python neste trabalho, utilize apenas a biblioteca *psycopg2*. Não será permitida a utilização de qualquer outra biblioteca de interação com BD, particularmente a biblioteca *Pandas*.

#### 2.2 Criação de Tabelas

Escreva um script em Python para criar as seguintes tabelas no seu banco de dados no servidor PostgreSQL da disciplina:

- Embarcacoes (id\_emb INT, nome VARCHAR, tipo VARCHAR)
- Tripulantes (id\_trp INT, nome VARCHAR, data\_nasc DATETIME, funcao VARCHAR, id\_emb INT)
- Empregados (id\_emp INT, nome VARCHAR, data\_nasc DATETIME, funcao VARCHAR)
- Movimentação (id\_mov INT, data DATETIME, tipo VARCHAR, id\_emb INT)
- Movimentacao\_Empregados (id\_mov INT, id\_emp INT)

**Importante:** Esse esquema é um subconjunto simplificado do esquema utilizado no Trabalho Prático I. De tal forma, realize as adaptações necessárias para que o seu database se adéque ao novo esquema. Caso necessário, utilize `DROP TABLE IF EXISTS` no seu script para remover as tabelas criadas anteriormente, antes de criar as novas.

## 2.3 Inserção de dados

Escreva um novo script em Python para inserir os dados no banco de dados como descritos nas Tabelas 1 a 5 .

id_emb	nome	tipo
1	Navio1	Cargueiro
2	Navio2	Passageiro
3	Navio3	Petroleiro
4	Navio4	Cargueiro

Tabela 1: Embarcacoes

id_trp	nome	data_nasc	funcao	id_emb
1	Tripulante1	1990-01-15	Oficial de Convés	1
2	Tripulante2	1992-03-20	Engenheiro	1
3	Tripulante3	1988-11-05	Comissário de Bordo	2
4	Tripulante4	1995-06-30	Oficial de Convés	3
5	Tripulante5	1991-07-10	Capitão	4
6	Tripulante6	1994-09-25	Engenheiro	4

Tabela 2: Tripulantes

id_emp	nome	data_nasc	funcao
1	Employee1	1985-05-12	Manutenção
2	Employee2	1993-02-28	Segurança
3	Employee3	1987-09-18	Logística
4	Employee4	1990-12-05	Limpeza
5	Employee5	2001-08-30	Manutenção

Tabela 3: Empregados

id_mov	data	tipo	id_emb
1	2023-09-01	Carga	1
2	2023-09-02	Embarque de Passageiros	2
3	2023-10-03	Abastecimento	3
4	2023-10-05	Descarga	1
5	2023-10-05	Manutenção	4

Tabela 4: Movimentações

id_mov	id_emp
1	1
1	3
2	2
3	1
3	4
4	1
4	3
5	1

Tabela 5: Movimentacoes\_Empregados

## 2.4 Consulta

Escreva um novo script em Python que realize as seguintes consultas:

- Retorne todas as embarcações e o número de tripulantes de cada embarcação.
- Retorne os Empregados envolvidos na movimentação de ID 1.
- Retorne a quantidade de movimentações que envolvem embarcações do tipo “Cargueiro”.

## 2.5 Transação

Escreva um novo script em Python que realize as seguintes operações em uma única transação:

1. Inserir nova Movimentação: (6, '2023-10-05', 'Manutenção', 1)
2. Inserir nova Movimentação\_Empregado: (6, 1)
3. Retornar a quantidade de movimentações que envolvem embarcações do tipo “Cargueiro”.

## 2.6 Procedimento Armazenado (*Stored Procedure*)

1. Crie um procedimento armazenado no banco de dados com nome **empregado\_do\_mes** que recebe como parâmetro uma data e retorna o id e nome do empregado que participou de mais movimentações naquela combinação Ano/Mês;
2. Escreva um novo script Python que chama o procedimento armazenado criado no item anterior passando como parâmetro a data 2023-10-01.

## 2.7 Gatilhos (*Triggers*)

1. Implemente um gatilho no banco de dados que dispara toda vez que um Tripulante é cadastrado ou quando o atributo “funcao” tem seu valor modificado. O gatilho deve garantir que somente um dos tripulantes tenha a função “Capitão”.
2. Crie um segundo gatilho que restrinja que somente empregados da manutenção possam ser escolhidos para executar movimentações do tipo “Manutenção”.
3. Por fim escreva um *script* em python que tenta inserir as tuplas como descrito na Tabela 6

id_mov	id_emp
5	5
5	2

Tabela 6: Tuplas na Tabela “Movimentacoes\_Empregados”

4. Adicione ao *script* uma linha que tenta modificar o valor do atributo “funcao” do Tripulante3 para “Capitão” segundo a Tabela 7.

id_trp	nome	data_nasc	funcao	id_emb
7	Tripulante7	1980-09-04	Capitão	4
8	Tripulante8	1985-03-03	Capitão	2

Tabela 7: Tuplas na Tabela “Tripulantes”

## 3 Entrega

- **Entrega até 23:59 do dia 16/11/2023, quinta, pelo *Classroom* da disciplina. Não serão aceitos envios posteriores.**
- Os trabalhos serão realizados pelas mesmas duplas do Trabalho I.
- O trabalho deve ser feito usando múltiplos scripts. Deve ser enviado um arquivo *.zip* contendo os scripts produzidos. O nome do arquivo deve seguir o seguinte formato: FBD\_Trabalho\_II\_Nome\_Aluno1\_Nome\_Aluno2.zip.
- **Os trabalhos serão apresentados no LEC/Computação na sexta, 17/11, no horário da aula com a presença dos dois membros da equipe para arguição.** Os alunos devem apresentar o trabalho enviado no *Classroom*. A apresentação é obrigatória e compõe a nota do trabalho.
- Qualquer dúvida, podem saná-las com os monitores responsáveis pelo trabalho através do email (Ítalo Abreu - italo.abreu@lsbd.ufc.br, Rafael Sá - rafael.sa@lsbd.ufc.br) ou pelo *google classroom*.

## 4 Critérios de Avaliação

- Entrega do arquivo *zip* conforme o modelo requisitado: 1 score.
- Script de Criação de Tabelas - 1 score.
- Script de População de Tabelas - 1 score.
- Script de Consultas - 1 score.
- Script de uso de Transações - 2 scores.
- Criação e Script de Procedimento Armazenado - 2 scores.
- Criação e Script de Gatilhos - 2 scores.
- Total: 10 scores.

Em cada script, a pontuação levará em consideração o seguinte:

- Corretamente abre, usa e fecha conexão com banco de dados.
- Corretamente abre, usa e fecha *cursor*, quando necessário.
- Atende aos objetivos propostos.