

CK0114 - Fundamentos de Bancos de Dados - 2025

Javam Machado, Lucas Sena, Lucas Cabral

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 *psycopg3*:

```
pip install psycopg3
```

Acesse a documentação do *psycopg3* 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*

Importante: Para interação com o banco de dados via Python neste trabalho, utilize apenas a biblioteca *psycopg3*. 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:

- Curso (id INT PRIMARY KEY, nome VARCHAR(100), regime VARCHAR(20), duracao INT)
- Aluno (id INT PRIMARY KEY, nome VARCHAR(100), curso_id INT REFERENCES Curso(id), semestre INT)
- Professor (id INT PRIMARY KEY, nome VARCHAR(100), area_especializacao VARCHAR(100), contato VARCHAR(100), curso_id INT REFERENCES Curso(id))
- Disciplina (id INT PRIMARY KEY, codigo VARCHAR(10) UNIQUE, nome VARCHAR(100), area_especializacao VARCHAR(100), carga_horaria INT, curso_id INT REFERENCES Curso(id))
- Turma (id INT PRIMARY KEY, codigo VARCHAR(10) UNIQUE, disciplina_id INT REFERENCES Disciplina(id), semestre VARCHAR(20), capacidade_maxima INT, estado VARCHAR(20), prof_id INT REFERENCES Professor(id))

- Aluno_Turma (aluno_id INT REFERENCES Aluno(id), turma_id INT REFERENCES Turma(id)), PRIMARY KEY (aluno_id, turma_id)

Importante: Esse esquema é um subconjunto simplificado do esquema utilizado no Trabalho Prático I. Assim, realize as adaptações necessárias para que o seu banco de dados fique conforme o 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 2 a 6 .

Curso

id	nome	regime	duracao
1	Ciências da Computação	Semestral	8
2	Engenharia de Software	Anual	10
3	Sistemas de Informação	Semestral	8

Tabela 1: Dados da tabela Curso

Aluno

id	nome	curso	semestre
1	João Silva	1	1
2	Maria Costa	1	1
3	Ana Souza	3	5
4	Pedro Almeida	2	3
5	Lucas Santos	2	3

Tabela 2: Dados da tabela Aluno

Professor

id	nome	area_especializacao	contato	curso_id
1	Maria Oliveira	Banco de Dados	maria@ufc.br	1
2	João Pereira	Redes de Computadores	joao@ufc.br	2
3	Ana Silva	Inteligência Artificial	ana@ufc.br	3
4	Paulo Santos	Engenharia de Software	paulo@ufc.br	2
5	Carla Mendes	Redes de Computadores	carla@ufc.br	1

Tabela 3: Dados da tabela Professor

Disciplina

id	codigo	nome	area_especializacao	carga_horaria	curso_id
1	BD001	Fundamentos de Bancos de Dados	Banco de Dados	60	1
2	IA002	Inteligência Computacional Aplicada	Inteligência Artificial	80	3
3	RS003	Segurança da Informação	Redes de Computadores	40	2
4	BD004	Introdução a Ciência de Dados	Banco de Dados	60	1
5	ES005	Qualidade de Software	Engenharia de Software	50	2

Tabela 4: Dados da tabela Disciplina

Turma

id	codigo	disciplina_id	semestre	capacidade_maxima	estado	prof_id
1	CC2024BD1	1	2024.2	4	Aberta	1
2	CC2024IA1	2	2024.2	4	Aberta	3
3	CC2024RS1	3	2024.1	8	Aberta	2
4	CC2024DS1	4	2024.2	4	Aberta	1
5	CC2024ES1	5	2024.2	8	Aberta	4

Tabela 5: Dados da tabela Turma

Aluno_Turma

aluno_id	turma_id
1	1
2	1
3	2
4	3
5	4
1	5
2	4
3	5
4	2
5	3

Tabela 6: Dados da tabela Aluno_Turmas

2.4 Consulta

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

- Retorne todas as turmas e a quantidade de alunos participantes de cada turma.
- Retorne os alunos matriculados na disciplina de “Fundamentos de Bancos de Dados”.
- Retorne a quantidade de professores do curso de “Ciências da Computação”.

2.5 Transação

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

1. Atualize o estado da turma “CC2024DS1” para “Fechado”.
2. Remova todas as matrículas de alunos na turma “CC2024DS1” na tabela Aluno_Turma.

2.6 Procedimento Armazenado (*Stored Procedure*)

1. Usando PL/pgSQL, crie um procedimento armazenado no banco de dados de nome **inc_semestre** que recebe como parâmetro um semestre e para todos os alunos que estão naquele semestre, o semestre é incrementado em 1;
2. Escreva um *script* Python que chama o procedimento armazenado criado no item anterior passando como parâmetro o valor 1 (*i.e.*, quem está no primeiro semestre agora vai para o segundo).

2.7 Gatilhos (*Triggers*)

1. Usando PL/pgSQL, implemente um gatilho no banco de dados que dispara toda vez que um aluno é adicionado a uma turma. O gatilho deve garantir que o número de alunos matriculados em uma turma não exceda a capacidade máxima da mesma.
2. Usando PL/pgSQL, crie um segundo gatilho que restrinja um aluno a não cursar mais do que 4 disciplinas em um semestre.

3. Escreva um *script* em Python que tenta inserir as tuplas na Tabela “Aluno_Turma” como descrito na Tabela 7

id_aluno	id_turma
3	1
5	1
4	1

Tabela 7: Tuplas na Tabela “Aluno_Turma”

4. Escreva um *script* em Python que tenta inserir as tuplas na Tabela “Aluno_Turma” como descrito na Tabela 8

id_aluno	id_turma
1	2
1	3
1	4

Tabela 8: Tuplas na Tabela “Aluno_Turma”

3 Entrega

- **Entrega até 23:59 do dia 30/01/2025, 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* Python. Os gatilhos e procedimento armazenado devem ser salvos em arquivos *.sql*. Deve ser enviado um único arquivo *.zip* contendo os *scripts* Python e os códigos PL/pgSQL 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, 31/01, 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, preferencialmente pelo *Google Classroom* ou por e-mail (Lucas Sena - lucas.sena@lsbd.ufc.br, Lucas Cabral - lucas.cabral@lsbd.ufc.br). Ao tirar dúvidas com os monitores por e-mail, coloque sempre ambos os monitores em cópia.

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.
- Código PL/pgSQL e *Script* de Procedimento Armazenado - 2 scores.
- Código PL/pgSQL e *Scripts* 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.