

Trabalho Prático I

Introdução a Banco de Dados

1 Introdução

Este trabalho tem como objetivo a familiarização com bancos de dados relacionais, o ambiente dos SGBDs e a realização de consultas utilizando a linguagem SQL (Structure Query Language), a principal linguagem de consulta a SGBDs relacionais. Conforme visto em aula, SQL possui comandos para criar tabelas, inserir, modificar e acessar dados. Alguns exemplos de SGBDs relacionais incluem Oracle, SQL Server, MySQL, PostgreSQL, DB2 e SQLite.

O banco de dados a ser utilizado neste trabalho contém dados sobre algumas das despesas públicas divulgadas no site da transparência do Governo Federal.¹

2 Instruções

Para realizar este trabalho prático, as seguintes ações são necessárias.

1. Instalar o Jupyter Notebooks (através do pacote Anaconda), que será utilizado no trabalho (detalhes na Seção 6).
2. Acessar o Moodle e fazer o download da base de dados de despesas públicas (arquivo `despesas_publicas_tp1.sql`) que deverá ser importada diretamente no Jupyter Notebook, utilizando o SQLite (`sqlite3`).
3. Entregar, via Moodle, um pacote **zip** contendo dois arquivos:
 - (a) um CSV com o nome `seu_numero_de_matricula.csv` (detalhes na Seção 3).
 - (b) um arquivo Jupyter Notebook (`.ipynb`) com as consultas realizadas utilizando o SQLite e possíveis visualizações para os dados. Poderão ser atribuídos pontos extras pela criatividade na visualização das consultas.
4. Para as consultas da parte 2 (Seção 5.2), indique nas respostas impressas o número da consulta correspondente.

#questão | #linhas | #colunas | #SQL

3 Formatação

O arquivo CSV gerado para submissão deverá conter **quatro colunas** separadas por “|”, onde cada linha segue o formato:
onde,

- #questão é o número da questão (veja Seção 5);
- #linhas é número de linhas da tabela resultante da consulta SQL;
- #colunas é número de colunas da tabela resultante da consulta SQL;
- SQL é o comando SQL select executado.

Exemplo: arquivo com nome 2015041962.csv contendo as seguintes linhas:

```
1|10|6|SELECT * FROM PROGRAMA_GOVERNO LIMIT 10;  
2|282|4|SELECT * FROM ORGAO WHERE NOME_ORGAO = 'Presidência da República';
```

4 Esquema Relacional

O banco de dados a ser criado e consultado possui o seguinte esquema relacional.

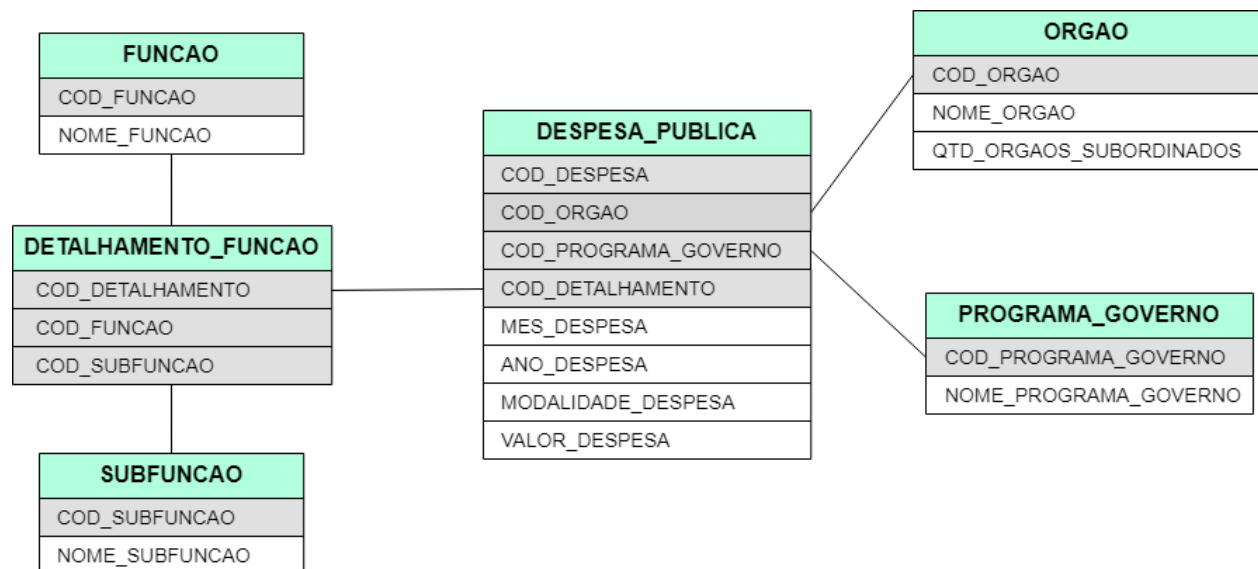


Figura 1: Esquema do banco de dados de despesas públicas

ORGAO(COD_ORGAO, NOME_ORGAO, QTD_ORGAOS_SUBORDINADOS)

¹Fonte dos dados: <http://www.portaldatransparencia.gov.br/download-de-dados/>

FUNCAO(COD_FUNCAO, NOME_FUNCAO)

SUBFUNCAO(COD_SUBFUNCAO, NOME_SUBFUNCAO)

DETALHAMENTO_FUNCAO(COD_DETALHAMENTO, COD_FUNCAO, COD_SUBFUNCAO)
DETALHAMENTO_FUNCAO[COD_FUNCAO] REFERENCIA FUNCAO[COD_FUNCAO]
DETALHAMENTO_FUNCAO[COD_SUBFUNCAO] REFERENCIA FUNCAO[COD_SUBFUNCAO]

PROGRAMA_GOVERNO(COD_PROGRAMA_GOVERNO, NOME_PROGRAMA_GOVERNO)

DESPESA_PUBLICA(COD_DESPESA, COD_ORGAO, COD_PROGRAMA_GOVERNO, COD_DETALHAMENTO,
MES_DESPESA, ANO_DESPESA, MODALIDADE_DESPESA, VALOR_DESPESA)
DESPESA_PUBLICA[COD_ORGAO] REFERENCIA ORGAO[COD_ORGAO]
DESPESA_PUBLICA[COD_SUBFUNCAO] REFERENCIA SUBFUNCAO[COD_SUFUNCAO]
DESPESA_PUBLICA[COD_PROGRAMA_GOVERNO] REFERENCIA
PROGRAMA_GOVERNO[COD_PROGRAMA_GOVERNO]

5 Consultas

O arquivo CSV gerado para submissão deverá conter exatamente **10 (dez) consultas SQL**, sendo **seis** consultas selecionadas da Seção 5.1 e **quatro** consultas da Seção 5.2.

5.1 Comandos SQL

Após criar e popular o banco de dados localmente, formule e execute **seis** das consultas especificadas a seguir. As consultas a serem formuladas e executadas serão em função do seu número de matrícula. Para isso, some 1 a cada um dos 6 últimos dígitos do seu número de matrícula. Os números resultantes indicam as consultas que você deve formular e executar. Caso seja gerado algum número repetido, considere a consulta posterior mais próxima ainda não selecionada. Por exemplo, se o seu número de matrícula fosse 2015**041962**, as suas consultas seriam as de número **1** ($0 + 1$), **5** ($4 + 1$), **2** ($1 + 1$), **10** ($9 + 1$), **7** ($6 + 1$) e **3** ($2 + 1$).

1. Liste todos os códigos e nomes de funções da subfunção que possui o maior número de funções.
2. Liste a média de despesas públicas por função (nome) e subfunção (nome), apenas nos casos em que a média excedeu mil reais.
3. Liste todos os nomes dos programas de governo e o valor das despesas totais no ano de 2018 para os mesmos, ordenando de forma decrescente pelo valor.
4. Liste o mês e ano, modalidade, programa de governo e valor de todas as despesas do órgão *Ministério da Educação* no primeiro semestre de 2018.

5. Liste o código e nome dos órgãos que possuíram os 5 maiores valores de despesas totais nas funções de *educação* e *saúde* durante o ano de 2018.
6. Liste todos os nomes de funções e subfunções das despesas públicas dos últimos 5 meses de 2018 e seus respectivos valores totais, ordenando de forma decrescente pelo valor.
7. Liste os programas de governo e a quantidade de órgãos governamentais distintos envolvidos nas despesas públicas desses programas. Nesta consulta, desconsidere valores de despesa negativos ou iguais a zero no período.
8. Liste os nomes das subfunções das despesas públicas envolvidas para cada órgão governamental, apenas nos casos em que o valor total das despesas excedeu 10 mil reais. Sua consulta deve retornar o nome do órgão, a subfunção da despesa, a quantidade de entradas de despesas públicas e o valor total dessas despesas.
9. Liste todas as despesas públicas (código da despesa, mês e ano da despesa, modalidade, valor) em que o órgão responsável é aquele que possui um número de órgãos subordinados maior do que a média.
10. Liste todas as despesas públicas (código da despesa, mês e ano da despesa, modalidade, valor), ordenadas de forma decrescente pelo valor, em que o programa de governo é aquele que possui o maior valor total em despesas durante o ano de 2018.
11. Liste os códigos e nomes dos órgãos governamentais relacionados ao programa de governo (ou programas, caso haja empate) que alcançou mais registros de despesas públicas durante o ano.

5.2 Tradução de Comandos da Álgebra Relacional

Além das seis consultas anteriores, traduza as seguintes expressões da álgebra relacional para o SQL e execute-as também.

12. $\pi_{\text{NOME_FUNCAO}, \text{VALOR_DESPESA}}(\text{FUNCAO} \bowtie \text{DETALHAMENTO_FUNCAO} \bowtie \text{DESPESA_PUBLICA})$
13. $\pi_{\text{MODALIDADE_DESPESA}, \text{VALOR_DESPESA}, \text{NOME_ORGAO}}(\text{ORGAO} \bowtie \sigma_{\text{MODALIDADE_DESPESA}='Reserva de Contingência'}(\text{DESPESA_PUBLICA}))$
14. $\pi_{\text{VALOR_DESPESA}, \text{NOME_PROGRAMA_GOVERNO}}(\text{DESPESA_PUBLICA} \bowtie \sigma_{\text{NOME_PROGRAMA_GOVERNO}='ENEM'}(\text{PROGRAMA_GOVERNO}))$
15. $\pi_{\text{NOME_PROGRAMA_GOVERNO}, \text{QTD_ORGAOS_SUBORDINADOS}, \text{VALOR_DESPESA}}(\text{DESPESA_PUBLICA} \bowtie \sigma_{\text{NOME_ORGAO}='Ministério da Educação'}(\text{ORGAO}) \bowtie \text{PROGRAMA_GOVERNO})$

6 Tutorial

A criação das consultas pode ser realizada através de um notebook Jupyter, instalado juntamente com o pacote Anaconda além de outras ferramentas que também acompanham a distribuição.

6.1 Download e instalação

Acesse a página de download da distribuição Anaconda² e selecione seu sistema operacional e a versão do Python que será instalada (recomendamos que seja escolhida a versão 3.7).

6.1.1 Ubuntu

1. Abra o terminal e execute o arquivo `.sh` baixado do site do Anaconda (basta abrir a pasta onde o pacote foi baixado e arrastá-lo até uma janela do terminal).
2. Após instalado, digite o comando `jupyter notebook` no terminal e uma janela do navegador será aberta com o Jupyter instanciado na pasta corrente do terminal.

```
$ jupyter notebook
```

6.1.2 macOS

1. O arquivo baixado no site é um pacote instalável do tipo `*.pkg`. Basta clicar duas vezes no arquivo para abri-lo e seguir os passos de instalação.

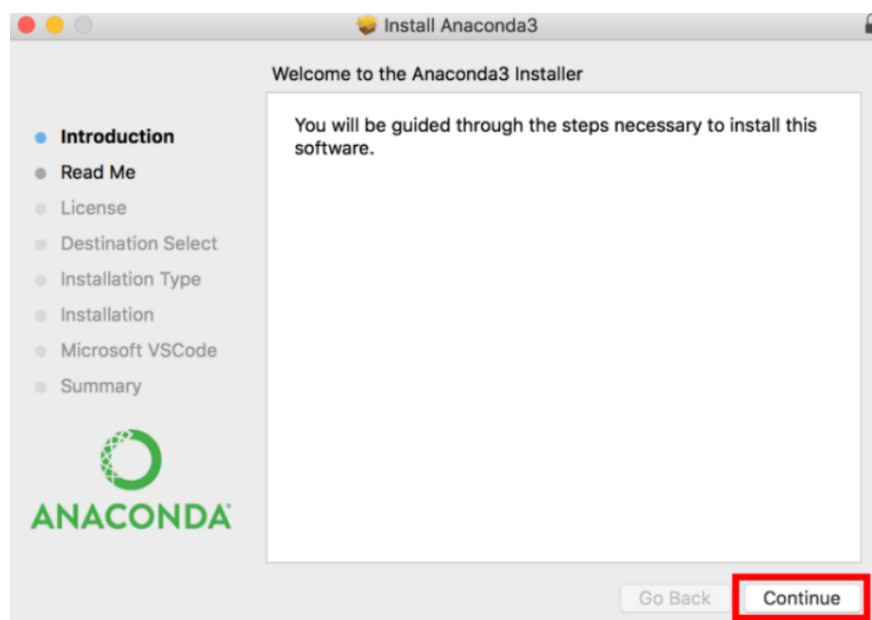


Figura 2: Instalação do pacote Anaconda no macOS

2. Após a instalação, o Jupyter pode ser aberto pela plataforma do Anaconda em “Apps” ou via linha de comando em um terminal, de forma semelhante ao Ubuntu.

6.1.3 Windows

1. O arquivo baixado no site é executável (`*.exe`). Basta clicar duas vezes no arquivo para abrir e seguir os passos da instalação. Apesar da recomendação do instalador por

²Distribuição Anaconda: <https://www.anaconda.com/distribution/>

não adicionar o pacote à variável PATH do Windows, o mais simples é adicioná-la, principalmente se você utilizar algum outro terminal que não seja o cmd padrão do Windows (Figura 3).

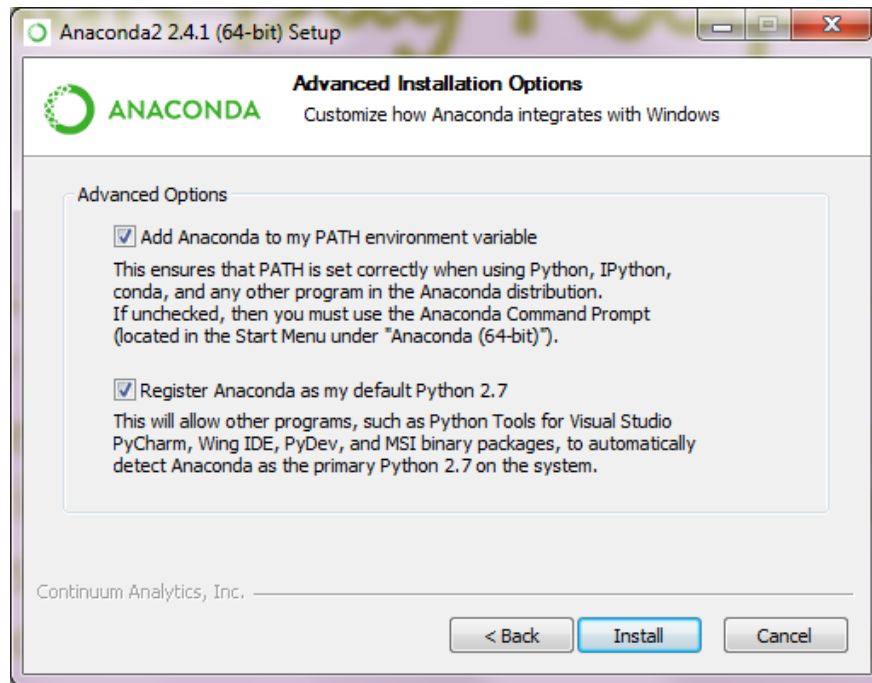


Figura 3: Instalação do pacote Anaconda no Windows

2. Após a instalação, o Jupyter pode ser aberto pela plataforma do Anaconda no “Menu Iniciar” do Windows.

6.2 Executando o Jupyter Notebook

Após a instalação, o Jupyter pode ser aberto, pela plataforma do Anaconda (“Menu Iniciar” no Windows, “Apps” no macOS/Ubuntu), ou via linha de comando em um terminal: `jupyter notebook`.

1. Ao abrir o Jupyter, a janela do navegador será direcionada para a pasta do usuário ou para a pasta de onde o comando foi executado, caso o Jupyter tenha sido aberto a partir do terminal (Figura 4).
2. É possível navegar entre os diretórios e criar pastas para armazenar seus arquivos notebook.
3. Para criar um novo notebook, basta clicar em “**New**” no canto superior direito e selecionar a linguagem do novo notebook que será criado. No nosso caso, Python 3.
4. Após a criação do notebook, a conexão com o banco de dados e execução das consultas será realizada conforme descrito na Seção 7.



Figura 4: Página inicial do Jupyter

7 SQLite

O SQLite é uma engine de bancos de dados autocontida. Para Python, existe a biblioteca `sqlite3` que permite a criação de bancos de dados e execução de consultas SQL sem nenhuma dependência adicional.

Existe um notebook de exemplo para a criação do banco, carga dos dados e execução de algumas consultas que pode ser utilizado como modelo para a realização do trabalho prático. O notebook está disponível no Github³.

Para a execução dos comandos do banco de dados e importação do arquivo `.sql`, serão necessários os imports das seguintes bibliotecas do Python:

```
import io
import sqlite3
import pandas as pd
```

7.1 Criação da conexão

No momento da criação da conexão, é informado o nome do arquivo onde será montado o banco de dados. No nosso caso, este arquivo não precisa ser pré-existente, porque iremos carregar as informações de um dump do banco de despesas públicas.

```
conn = sqlite3.connect('tp1.db')
cursor = conn.cursor()
```

O cursor será utilizado para executar o comando de criação do banco de dados a partir do arquivo `.sql`. Já a conexão, será utilizada para executar as consultas.

³Notebook modelo para o TP: https://github.com/naterciaaguilar/jupyter_intro_class_ibd/blob/master/notebooks/3.%20Jupyter%20e%20SQLite.ipynb

7.2 Carga dos dados

A carga dos dados deve ser realizada apenas uma vez. Então, quando iniciar a realização do TP, você deve executar essa parte, mas quando for utilizar o banco de dados novamente, só é necessário conectar ao banco (Seção 7.1) e realizar as consultas (Seção 7.3), sem passar por esta etapa.

```
f = io.open('despesas_publicas_tp1.sql', 'r')
sql = f.read()
cursor.executescript(sql)
```

7.3 Execução das consultas

Com o banco de dados criado, podemos executar consultas de forma simples, recuperando seu resultado em um *DataFrame* utilizando o seguinte comando:

```
df = pd.read_sql_query("SELECT * "+
                        "FROM ORGAO "+
                        "ORDER BY QTD_ORGAOS_SUBORDINADOS DESC",
                        conn)
```

A execução dessa consulta, salvará o resultado da mesma na variável `df`, que pode ser visualizada no Jupyter Notebook da seguinte maneira:

| | COD_ORGAO | NOME_ORGAO | QTD_ORGAOS_SUBORDINADOS |
|----|-----------|---|-------------------------|
| 0 | 26000 | Ministério da Educação | 104 |
| 1 | 30000 | Ministério da Justiça e Segurança Pública | 6 |
| 2 | 22000 | Ministério da Agricultura, Pecuária e Abastec | 5 |
| 3 | 24000 | Ministério da Ciência, Tecnologia, Inovações | 4 |
| 4 | 52000 | Ministério da Defesa | 4 |
| 5 | 53000 | Ministério da Integração Nacional | 4 |
| 6 | 25000 | Ministério da Fazenda | 3 |
| 7 | 36000 | Ministério da Saúde | 3 |
| 8 | 20000 | Presidência da República | 2 |
| 9 | 39000 | Ministério dos Transportes, Portos e Aviação | 2 |
| 10 | 44000 | Ministério do Meio Ambiente | 2 |
| 11 | 55000 | Ministério do Desenvolvimento Social | 2 |
| 12 | 35000 | Ministério das Relações Exteriores | 1 |
| 13 | 81000 | Ministério dos Direitos Humanos | 1 |

Figura 5: Consulta realizada no notebook a partir do SQLite e Pandas