Bases de dados

Nos tempos atuais, convivemos com a necessidade constante de interação e manipulação de dados.

Chamamos de *dado* qualquer elemento que possa ser processado por um computador (conjunto de números, strings, imagens, senhas, etc.).

- Hoje em dia dados são coletados a todo momento: temperatura diária, fotos de radares, retirada e devolução de livros de uma biblioteca, códigos de barra dos produtos comprados no supermercado, etc.

Chamamos de *informação* a interpretação que damos a um conjunto de dados, tornando-os significativos para algum contexto.

- Os dados são coletados para algum propósito: sabendo a média de temperatura de um local e as variações podemos tentar fazer previsões, se há muita multa sendo aplicada em um certo local podemos decidir por colocar um semáforo, um usuário pode fazer uma busca no acervo da biblioteca sabendo qual o estado do livro que deseja, o supermercado pode ter controle do estoque atual de cada produto e receber alertas automáticos se for necessária a reposição, etc.

Em termos gerais, o computador só é capaz de manipular dados e estes se transformam em informações quando interpretados por seus usuários.

Usamos o termo *base de dados* (ou *banco de dados*) para indicar um conjunto de dados sobre o qual é possível realizar:

1. **armazenamento**: os dados precisam ser preservados para futura utilização
2. **recuperação**: os dados já cadastrados precisam estar à disposição para consulta
3. **visualização**: os dados recuperados precisam ser exibidos para o usuário

Normalmente, bases de dados são usadas para problemas grandes e complexos como controlar a lista de clientes de uma empresa, de correntistas de um banco, de alunos de uma universidade, etc.

Em muitos casos mais complexos, um *sistema gerenciador de bancos de dados*, que é um software especializado para criação e manipulação de bancos de dados, é utilizado. Em disciplinas mais avançadas da Ciência da Computação você poderá estudar esse tipo de sistema, que também se preocupa com segurança e consistência dos dados.

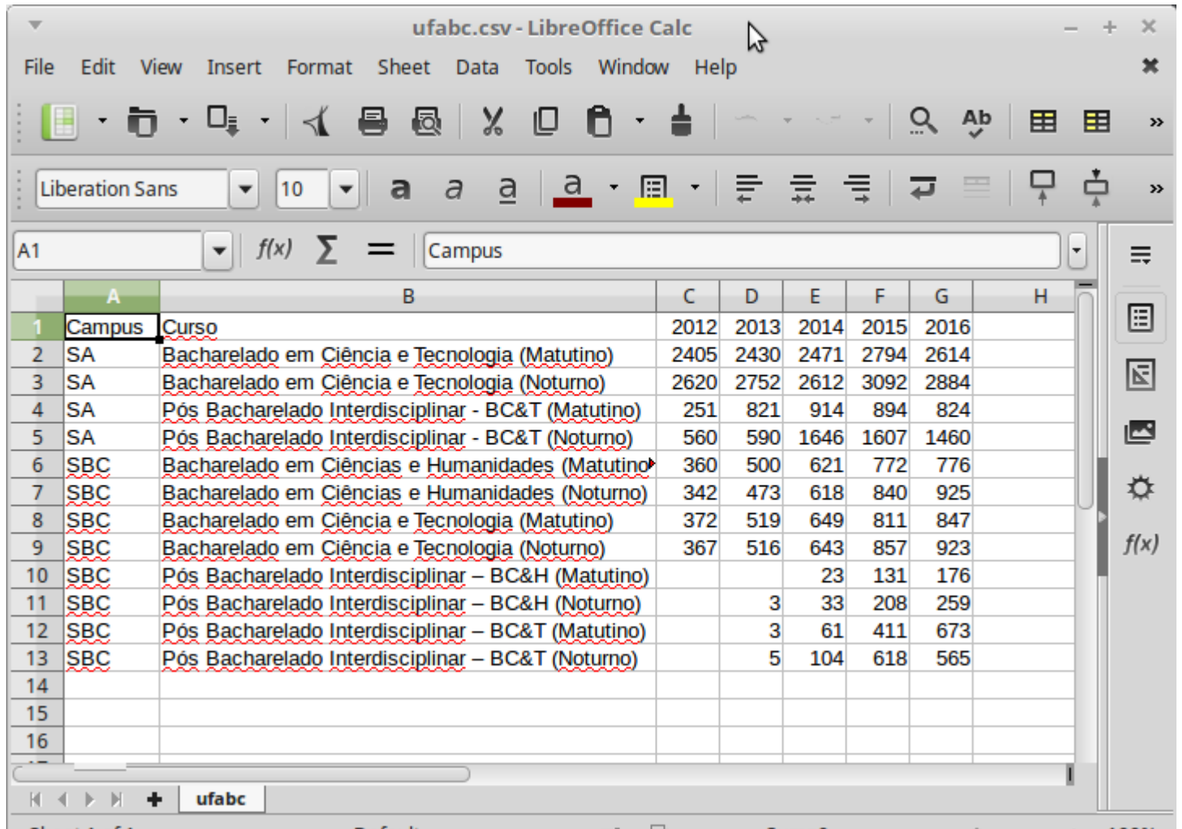
Nessa aula veremos como manipular uma base de dados de forma mais simples, que lhe será útil independente do caminho que você seguir.

> **De modo algum iremos cobrir nessa aula (ou nessa disciplina) tudo que existe sobre manipulação de bases/bancos de dados.**

Organização de dados com planilha

Planilhas são formas de representação de uma base de dados por meio de *matrizes* ou *tabelas*.

Por exemplo, se a nossa base armazena a quantidade de alunos matriculados na UFABC nos últimos anos por curso, podemos utilizar uma planilha onde as colunas indicam o ano e cada linha armazena o nome do curso e a quantidade de alunos por ano:



	A	B	C	D	E	F	G	H
1	Campus	Curso	2012	2013	2014	2015	2016	
2	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405	2430	2471	2794	2614	
3	SA	Bacharelado em Ciência e Tecnologia (Noturno)	2620	2752	2612	3092	2884	
4	SA	Pós Bacharelado Interdisciplinar - BC&T (Matutino)	251	821	914	894	824	
5	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560	590	1646	1607	1460	
6	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360	500	621	772	776	
7	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342	473	618	840	925	
8	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372	519	649	811	847	
9	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367	516	643	857	923	
10	SBC	Pós Bacharelado Interdisciplinar – BC&H (Matutino)			23	131	176	
11	SBC	Pós Bacharelado Interdisciplinar – BC&H (Noturno)		3	33	208	259	
12	SBC	Pós Bacharelado Interdisciplinar – BC&T (Matutino)		3	61	411	673	
13	SBC	Pós Bacharelado Interdisciplinar – BC&T (Noturno)		5	104	618	565	
14								
15								
16								

Esse tipo de organização facilita a visualização dos dados e permite a identificação precisa de um item por meio de sua linha e coluna.

Quando usamos tabelas como base de dados, é importante manter a **consistência** dos dados: cada coluna tem um único rótulo, os rótulos ficam no topo da tabela e não pode haver linha em branco entre os rótulos e os dados.

Também é importante manter a tabela **válida**: uma coluna contém elementos do mesmo tipo (números, textos, data, hora, moeda, etc.).

Isso é muito **importante**: um computador consegue fazer cálculos em cima de números ou mesmo comparar textos entre si, mas não consegue fazer muito se os tipos forem diferentes.

Arquivos estruturados

Arquivos estruturados são arquivos de dados que possuem uma estrutura rígida e pré-definida.

Eles são muito utilizados para troca de dados entre sistemas computacionais diferentes.

Nesses arquivos existem caracteres que farão papel de *delimitadores* de linhas e colunas.

Por exemplo, se o caractere `;` separa colunas e o caractere `quebra de linha` separa linhas, então a tabela mostrada na figura anterior pode ser armazenada da seguinte forma:

```
Campus;Curso;2012;2013;2014;2015;2016
SA;Bacharelado em Ciência e Tecnologia (Matutino);2405;2430;2471;2794;2614
SA;Bacharelado em Ciência e Tecnologia (Noturno);2620;2732;2612;3092;2884
SA;Pós Bacharelado Interdisciplinar - BC&T (Matutino);251;821;914;894;824
SA;Pós Bacharelado Interdisciplinar - BC&T (Noturno);560;590;1646;1607;1460
SBC;Bacharelado em Ciências e Humanidades (Matutino);360;500;621;772;776
SBC;Bacharelado em Ciências e Humanidades (Noturno);342;473;618;840;925
SBC;Bacharelado em Ciência e Tecnologia (Matutino);372;519;649;811;847
SBC;Bacharelado em Ciência e Tecnologia (Noturno);367;516;643;857;923
SBC;Pós Bacharelado Interdisciplinar – BC&H (Matutino);;;23;131;176
SBC;Pós Bacharelado Interdisciplinar – BC&H (Noturno);;3;33;208;259
SBC;Pós Bacharelado Interdisciplinar – BC&T (Matutino);;3;61;411;673
SBC;Pós Bacharelado Interdisciplinar – BC&T (Noturno);;5;104;618;565
```

Note que todas as linhas possuem seis separadores de coluna, indicando que a tabela possui sete colunas.

Esse formato de arquivo é conhecido como **CSV** (*Comma-separated values*).

> Qualquer programa que manipula planilhas (Excel, Libreoffice Calc, Planilhas Google) e mesmo programas simples como o bloco de notas conseguem salvar, abrir e manipular arquivos em formato CSV. Com os programas de planilhas, no entanto, é possível criar gráficos, atualizar informações, criar novas linhas e colunas, ordenar os dados de acordo com alguma das colunas, filtrar dados para que apenas linhas relevantes sejam mostradas, etc. Todas essas coisas e muitas outras podem ser feitas usando Python.

> **No restante dessa aula veremos como manipular bases de dados salvas em arquivos CSV utilizando Python.**

Manipulação de dados em Python

Em Python, uma *biblioteca* é simplesmente um conjunto de funções (vimos funções na última aula, lembra?).

Por padrão, várias bibliotecas já são inclusas, mas para realização de tarefas mais específicas precisamos incluir novas bibliotecas em nosso programa.

Pandas é uma biblioteca que fornece funções para manipulação e análise de dados em formato de tabela.

> **De modo algum conseguiremos cobrir todas as funcionalidades disponibilizadas pelo Pandas.**

Para utilizá-la, o começo do nosso programa deve conter a linha `import pandas as pd` .

In [2]:
```
import pandas as pd
```

Agora temos disponíveis as funções que estão definidas na biblioteca Pandas.

> **Antes de continuar, certifique-se de que os arquivos `ufabc.csv` e `notas.csv` foram baixados e estão no mesmo diretório deste notebook.**

Podemos, por exemplo, abrir o arquivo `ufabc.csv` por meio da função `pd.read_csv()` .

Esse arquivo contém informação sobre a quantidade de alunos matriculados em cada ano para cada curso ofertado pela UFABC.

Essa função recebe o nome do arquivo e o carectere que é utilizado como separador de colunas.

In [18]:
```
ufabc = pd.read_csv("ufabc.csv", sep=";")
display(ufabc)
print(ufabc)
```

	Campus	Curso	2012	2013	2014	2015	2016
0	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405.0	2430.0	2471	2794	2614
1	SA	Bacharelado em Ciência e Tecnologia (Noturno)	2620.0	2732.0	2612	3092	2884
2	SA	Pós Bacharelado Interdisciplinar - BC&T (Matut...	251.0	821.0	914	894	824
3	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560.0	590.0	1646	1607	1460
4	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360.0	500.0	621	772	776
5	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342.0	473.0	618	840	925
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
7	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367.0	516.0	643	857	923
8	SBC	Pós Bacharelado Interdisciplinar - BC&H (Matut...	NaN	NaN	23	131	176
9	SBC	Pós Bacharelado Interdisciplinar - BC&H (Noturno)	NaN	3.0	33	208	259
10	SBC	Pós Bacharelado Interdisciplinar - BC&T (Matut...	NaN	3.0	61	411	673

O resultado da chamada à função `pd.read_csv()` foi colocado na variável `ufabc`.

Quando damos um valor a uma variável em Python, podemos verificar qual é o tipo dela por meio da função `type()`.

In [6]:
```
a = 7
print(type(a)) # o tipo de a é inteiro
b = "oi"
print(type(b)) # o tipo de b é string
```

```
<class 'int'>
<class 'str'>
```

A variável `ufabc` é do tipo DataFrame.

In [5]:
```
print(type(ufabc))
```

```
<class 'pandas.core.frame.DataFrame'>
```

Um DataFrame é simplesmente uma estrutura de dados tabular.

Esse nome vai ser importante para diferenciarmos o que pode ser feito sobre esse tipo de variável.

Por exemplo, em variáveis do tipo inteiro, podemos fazer soma, subtração, multiplicação, etc.

Vamos ver o que é possível fazer em DataFrames.

Em primeiro lugar, nós podemos visualizar o conteúdo de um DataFrame.

Perceba como a primeira linha contém os rótulos das colunas.

In [7]:
```
print(ufabc)
```

```
   Campus                                          Curso    2012    2013  \
0      SA     Bacharelado em Ciência e Tecnologia (Matutino)  2405.0  2430.0
1      SA      Bacharelado em Ciência e Tecnologia (Noturno)  2620.0  2732.0
2      SA  Pós Bacharelado Interdisciplinar - BC&T (Matut...   251.0   821.0
3      SA  Pós Bacharelado Interdisciplinar - BC&T (Noturno)   560.0   590.0
4     SBC   Bacharelado em Ciências e Humanidades (Matutino)   360.0   500.0
5     SBC    Bacharelado em Ciências e Humanidades (Noturno)   342.0   473.0
6     SBC     Bacharelado em Ciência e Tecnologia (Matutino)   372.0   519.0
7     SBC      Bacharelado em Ciência e Tecnologia (Noturno)   367.0   516.0
8     SBC  Pós Bacharelado Interdisciplinar - BC&H (Matut...     NaN     NaN
9     SBC  Pós Bacharelado Interdisciplinar - BC&H (Noturno)     NaN     3.0
10    SBC  Pós Bacharelado Interdisciplinar - BC&T (Matut...     NaN     3.0
11    SBC  Pós Bacharelado Interdisciplinar - BC&T (Noturno)     NaN     5.0

    2014  2015  2016
0   2471  2794  2614
1   2612  3092  2884
2    914   894   824
3   1646  1607  1460
4    621   772   776
5    618   840   925
6    649   811   847
7    643   857   923
8     23   131   176
9     33   208   259
10    61   411   673
11   104   618   565
```

Podemos também visualizar apenas as primeiras linhas (o que vai ser mais útil em arquivos maiores):

In [8]:
```
print(ufabc.head(6)) # o parâmetro de head() indica o número de linhas impressas
```

```
  Campus                                          Curso    2012    2013  \
0     SA     Bacharelado em Ciência e Tecnologia (Matutino)  2405.0  2430.0
1     SA      Bacharelado em Ciência e Tecnologia (Noturno)  2620.0  2732.0
2     SA  Pós Bacharelado Interdisciplinar - BC&T (Matut...   251.0   821.0
3     SA  Pós Bacharelado Interdisciplinar - BC&T (Noturno)   560.0   590.0
4    SBC   Bacharelado em Ciências e Humanidades (Matutino)   360.0   500.0
5    SBC    Bacharelado em Ciências e Humanidades (Noturno)   342.0   473.0

   2014  2015  2016
0  2471  2794  2614
1  2612  3092  2884
2   914   894   824
3  1646  1607  1460
4   621   772   776
5   618   840   925
```

Podemos visualizar a tabela de um jeito visualmente mais agradável.

Para isso, vamos utilizar uma outra biblioteca, que fornece a função `display()` :

In [9]:
```
from IPython.display import display
display(ufabc)
display(ufabc.head(3))
```

	Campus	Curso	2012	2013	2014	2015	2016
0	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405.0	2430.0	2471	2794	2614
1	SA	Bacharelado em Ciência e Tecnologia (Noturno)	2620.0	2732.0	2612	3092	2884
2	SA	Pós Bacharelado Interdisciplinar - BC&T (Matut...	251.0	821.0	914	894	824
3	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560.0	590.0	1646	1607	1460
4	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360.0	500.0	621	772	776
5	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342.0	473.0	618	840	925
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
7	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367.0	516.0	643	857	923
8	SBC	Pós Bacharelado Interdisciplinar - BC&H (Matut...	NaN	NaN	23	131	176
9	SBC	Pós Bacharelado Interdisciplinar - BC&H (Noturno)	NaN	3.0	33	208	259
10	SBC	Pós Bacharelado Interdisciplinar - BC&T (Matut...	NaN	3.0	61	411	673

Faça você mesmo

Altere o programa a seguir para abrir o arquivo `notas.csv`, salvando-o na variável `notas` (assim como o arquivo `ufabc.csv` foi salvo na variável `ufabc`).

Em seguida peça para visualizá-lo.

In [29]:
```
notas = pd.read_csv("notas.csv", sep=";")
display(notas)
print(notas)
```

	RA	Aluno	Prova 1	Prova 2	Prova Sub	Listas
0	15	AA	0.7	6.0	NaN	5.8
1	16	AB	5.0	NaN	NaN	2.6
2	18	BA	4.3	2.7	NaN	4.6
3	45	CA	2.1	8.3	NaN	7.5
4	89	CD	8.6	9.5	NaN	9.5
5	34	CF	10.0	9.0	NaN	10.0
6	23	FT	1.4	10.0	NaN	0.9
7	11	FU	6.9	NaN	NaN	3.2
8	67	GA	NaN	4.2	NaN	0.0

```
   RA Aluno  Prova 1  Prova 2  Prova Sub  Listas
0  15    AA      0.7      6.0        NaN     5.8
1  16    AB      5.0      NaN        NaN     2.6
2  18    BA      4.3      2.7        NaN     4.6
3  45    CA      2.1      8.3        NaN     7.5
4  89    CD      8.6      9.5        NaN     9.5
5  34    CF     10.0      9.0        NaN    10.0
6  23    FT      1.4     10.0        NaN     0.9
7  11    FU      6.9      NaN        NaN     3.2
8  67    GA      NaN      4.2        NaN     0.0
```

Podemos verificar o tamanho do DataFrame (número de linhas, número de colunas):

In [20]:
```
print(ufabc.shape)
```

```
(12, 7)
```

Podemos visualizar uma coluna específica do DataFrame.

Basta indicar uma lista com o rótulo da coluna:

In [10]:
```
coluna = ["Curso"] # essa é uma lista que tem um elemento, o texto "Curso"
display(ufabc[coluna])
```

	Curso
0	Bacharelado em Ciência e Tecnologia (Matutino)
1	Bacharelado em Ciência e Tecnologia (Noturno)
2	Pós Bacharelado Interdisciplinar - BC&T (Matut...
3	Pós Bacharelado Interdisciplinar - BC&T (Noturno)
4	Bacharelado em Ciências e Humanidades (Matutino)
5	Bacharelado em Ciências e Humanidades (Noturno)
6	Bacharelado em Ciência e Tecnologia (Matutino)
7	Bacharelado em Ciência e Tecnologia (Noturno)
8	Pós Bacharelado Interdisciplinar - BC&H (Matut...
9	Pós Bacharelado Interdisciplinar - BC&H (Noturno)
10	Pós Bacharelado Interdisciplinar - BC&T (Matut...
11	Pós Bacharelado Interdisciplinar - BC&T (Noturno)

In [11]:
```
coluna = ["2015"]
display(ufabc[coluna])
```

	2015
0	2794
1	3092
2	894
3	1607
4	772
5	840
6	811
7	857
8	131
9	208
10	411
11	618

Ou visualizar várias colunas:

In [12]:
```
colunas = ["2012", "2014", "2016"] # essa lista contém 3 elementos
display(ufabc[colunas])
```

	2012	2014	2016
0	2405.0	2471	2614
1	2620.0	2612	2884
2	251.0	914	824
3	560.0	1646	1460
4	360.0	621	776
5	342.0	618	925
6	372.0	649	847
7	367.0	643	923
8	NaN	23	176
9	NaN	33	259
10	NaN	61	673

Note nos exemplos anteriores que a primeira coluna não tem rótulo e contém números de 0 a 11 que não estão presentes no arquivo original.

Essa coluna contém o que chamamos de *índice* da tabela: cada linha tem um único índice que a identifica. A primeira linha tem índice 0, a segunda tem índice 1 e assim por diante.

Podemos utilizar os índices para visualizar uma linha específica usando `loc` :

In [21]:
```
linha = [4] # essa é uma lista que tem um elemento, o número 4
display(ufabc.loc[linha])
```

	Campus	Curso	2012	2013	2014	2015	2016
4	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360.0	500.0	621	772	776

Ou visualizar várias linhas:

In [13]:
```
linhas = [0, 3, 6, 8] # essa lista contém 4 elementos
display(ufabc.loc[linhas])
```

	Campus	Curso	2012	2013	2014	2015	2016
0	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405.0	2430.0	2471	2794	2614
3	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560.0	590.0	1646	1607	1460
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
8	SBC	Pós Bacharelado Interdisciplinar - BC&H (Matut...	NaN	NaN	23	131	176

Podemos filtrar o conteúdo a ser impresso de acordo com alguma condição.

Por exemplo, queremos visualizar apenas as linhas dos cursos que tiveram mais de 800 alunos matriculados no ano de 2015:

In [14]:
```
condicao = ufabc["2015"] > 800
display(ufabc[condicao])
```

	Campus	Curso	2012	2013	2014	2015	2016
0	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405.0	2430.0	2471	2794	2614
1	SA	Bacharelado em Ciência e Tecnologia (Noturno)	2620.0	2732.0	2612	3092	2884
2	SA	Pós Bacharelado Interdisciplinar - BC&T (Matut...	251.0	821.0	914	894	824
3	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560.0	590.0	1646	1607	1460
5	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342.0	473.0	618	840	925
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
7	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367.0	516.0	643	857	923

Ou então apenas as linhas dos cursos de São Bernardo do Campo:

In [15]:
```
condicao = ufabc["Campus"] == "SBC"
display(ufabc[condicao])
```

	Campus	Curso	2012	2013	2014	2015	2016
4	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360.0	500.0	621	772	776
5	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342.0	473.0	618	840	925
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
7	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367.0	516.0	643	857	923
8	SBC	Pós Bacharelado Interdisciplinar - BC&H (Matut...	NaN	NaN	23	131	176
9	SBC	Pós Bacharelado Interdisciplinar - BC&H (Noturno)	NaN	3.0	33	208	259
10	SBC	Pós Bacharelado Interdisciplinar - BC&T (Matut...	NaN	3.0	61	411	673
11	SBC	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	NaN	5.0	104	618	565

Na verdade, seria interessante visualizar apenas os cursos que de fato atendem a essa condição.

Podemos então, dentro das linhas selecionadas, especificar a coluna (ou colunas) que queremos visualizar:

In [16]:
```
condicao = ufabc["Campus"] == "SBC"
display(ufabc[condicao]["Curso"])
```

```
4      Bacharelado em Ciências e Humanidades (Matutino)
5       Bacharelado em Ciências e Humanidades (Noturno)
6         Bacharelado em Ciência e Tecnologia (Matutino)
7          Bacharelado em Ciência e Tecnologia (Noturno)
8     Pós Bacharelado Interdisciplinar - BC&H (Matut...
9     Pós Bacharelado Interdisciplinar - BC&H (Noturno)
10    Pós Bacharelado Interdisciplinar - BC&T (Matut...
11    Pós Bacharelado Interdisciplinar - BC&T (Noturno)
Name: Curso, dtype: object
```

Faça você mesmo

Peça para visualizar apenas os alunos que tiveram nota maior do que 5.0 **em ambas as provas**.

In [23]:
```
condicao = (notas["Prova 1"] > 5) & (notas["Prova 2"] > 5)
display(notas[condicao])
```

	RA	Aluno	Prova 1	Prova 2	Prova Sub	Listas
4	89	CD	8.6	9.5	NaN	9.5
5	34	CF	10.0	9.0	NaN	10.0

Podemos ordenar toda a tabela com relação aos valores de uma coluna com a função `sort_values()` .

Por exemplo, queremos observar os cursos que tiveram mais alunos matriculados no ano de 2016:

In [24]:
```
display(ufabc.sort_values(by=["2016"], ascending=False))
```

	Campus	Curso	2012	2013	2014	2015	2016
1	SA	Bacharelado em Ciência e Tecnologia (Noturno)	2620.0	2732.0	2612	3092	2884
0	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405.0	2430.0	2471	2794	2614
3	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560.0	590.0	1646	1607	1460
5	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342.0	473.0	618	840	925
7	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367.0	516.0	643	857	923
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
2	SA	Pós Bacharelado Interdisciplinar - BC&T (Matut...	251.0	821.0	914	894	824
4	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360.0	500.0	621	772	776
10	SBC	Pós Bacharelado Interdisciplinar - BC&T (Matut...	NaN	3.0	61	411	673
11	SBC	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	NaN	5.0	104	618	565
9	SBC	Pós Bacharelado Interdisciplinar - BC&H (Noturno)	NaN	3.0	33	208	259

Mas note que isso foi apenas uma visualização.

A tabela original não mudou:

In [25]:
```
display(ufabc)
```

	Campus	Curso	2012	2013	2014	2015	2016
0	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405.0	2430.0	2471	2794	2614
1	SA	Bacharelado em Ciência e Tecnologia (Noturno)	2620.0	2732.0	2612	3092	2884
2	SA	Pós Bacharelado Interdisciplinar - BC&T (Matut...	251.0	821.0	914	894	824
3	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560.0	590.0	1646	1607	1460
4	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360.0	500.0	621	772	776
5	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342.0	473.0	618	840	925
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
7	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367.0	516.0	643	857	923
8	SBC	Pós Bacharelado Interdisciplinar - BC&H (Matut...	NaN	NaN	23	131	176
9	SBC	Pós Bacharelado Interdisciplinar - BC&H (Noturno)	NaN	3.0	33	208	259
10	SBC	Pós Bacharelado Interdisciplinar - BC&T (Matut...	NaN	3.0	61	411	673
11	SBC	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	NaN	5.0	104	618	565

Para de fato aplicar a ordenação na tabela, precisamos indicar isso na função `sort_values()` com o parâmetro `inplace` :

In [26]:
```
ufabc.sort_values(by=["2016"], ascending=False, inplace=True)
display(ufabc)
```

	Campus	Curso	2012	2013	2014	2015	2016
1	SA	Bacharelado em Ciência e Tecnologia (Noturno)	2620.0	2732.0	2612	3092	2884
0	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405.0	2430.0	2471	2794	2614
3	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560.0	590.0	1646	1607	1460
5	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342.0	473.0	618	840	925
7	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367.0	516.0	643	857	923
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
2	SA	Pós Bacharelado Interdisciplinar - BC&T (Matut...	251.0	821.0	914	894	824
4	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360.0	500.0	621	772	776
10	SBC	Pós Bacharelado Interdisciplinar - BC&T (Matut...	NaN	3.0	61	411	673
11	SBC	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	NaN	5.0	104	618	565
9	SBC	Pós Bacharelado Interdisciplinar - BC&H (Noturno)	NaN	3.0	33	208	259

A biblioteca Pandas oferece um jeito fácil de lidar com valores faltantes das tabelas.

Em 2012, a UFABC ainda não oferecia os cursos de formação específica no campus de São Bernardo do Campo. Se você visualizar o arquivo CSV em um programa de planilhas ou mesmo em um programa tipo bloco de notas (veja a primeira figura desse notebook), verá que alguns elementos da tabela estão vazios.

Como você pode ver nas visualizações até agora, o Pandas não deixa a célula vazia, mas sim a preenche com um valor especial, o `NaN` .

E ele fornece uma função chamada `fillna()` , que preenche as células vazias com um valor desejado.

No nosso exemplo, vamos preencher com o valor 0, já que de fato havia 0 alunos matriculados em SBC.

Novamente, é preciso especificar que a mudança tem que ser de fato gravada na tabela:

In [27]:
```
ufabc.fillna(0, inplace=True)
display(ufabc)
```

	Campus	Curso	2012	2013	2014	2015	2016
1	SA	Bacharelado em Ciência e Tecnologia (Noturno)	2620.0	2732.0	2612	3092	2884
0	SA	Bacharelado em Ciência e Tecnologia (Matutino)	2405.0	2430.0	2471	2794	2614
3	SA	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	560.0	590.0	1646	1607	1460
5	SBC	Bacharelado em Ciências e Humanidades (Noturno)	342.0	473.0	618	840	925
7	SBC	Bacharelado em Ciência e Tecnologia (Noturno)	367.0	516.0	643	857	923
6	SBC	Bacharelado em Ciência e Tecnologia (Matutino)	372.0	519.0	649	811	847
2	SA	Pós Bacharelado Interdisciplinar - BC&T (Matut...	251.0	821.0	914	894	824
4	SBC	Bacharelado em Ciências e Humanidades (Matutino)	360.0	500.0	621	772	776
10	SBC	Pós Bacharelado Interdisciplinar - BC&T (Matut...	0.0	3.0	61	411	673
11	SBC	Pós Bacharelado Interdisciplinar - BC&T (Noturno)	0.0	5.0	104	618	565
9	SBC	Pós Bacharelado Interdisciplinar - BC&H (Noturno)	0.0	3.0	33	208	259

Faça você mesmo

Preencha as notas faltantes do arquivo de notas.

In [28]:
```
notas.fillna(0, inplace=True)
display(notas)
```

	RA	Aluno	Prova 1	Prova 2	Prova Sub	Listas
0	15	AA	0.7	6.0	0.0	5.8
1	16	AB	5.0	0.0	0.0	2.6
2	18	BA	4.3	2.7	0.0	4.6
3	45	CA	2.1	8.3	0.0	7.5
4	89	CD	8.6	9.5	0.0	9.5
5	34	CF	10.0	9.0	0.0	10.0
6	23	FT	1.4	10.0	0.0	0.9
7	11	FU	6.9	0.0	0.0	3.2
8	67	GA	0.0	4.2	0.0	0.0

E se quisermos saber o total de alunos matriculados em um certo ano?

O código a seguir soma os valores da coluna 2016 para descobrir isso no campus de Santo André.

Ele utiliza o comando `at[]`, que recebe um índice seguido do nome da coluna e retorna o elemento que está armazenado naquela posição da tabela.

In [ ]:
```
# queremos apenas olhar para o campus de SA:
condicao = ufabc["Campus"] == "SA"
# criamos uma lista com os índices das linhas onde o campus é SA:
indices_SA = list(ufabc[condicao].index.values)
# a variável total_SA vai fazer a soma dos alunos matriculados:
total_SA = 0
for i in indices_SA:
    # para cada linha selecionada, queremos apenas o conteúdo da coluna 2016:
    total_SA = total_SA + ufabc.at[i, "2016"]

print("Total de alunos matriculados em Santo André em 2016:", total_SA)
```

Mas o Pandas oferece uma função `sum()` que faz esse cálculo para nós mais facilmente.

É importante perceber que isso só é possível porque as colunas estão *válidas*: exceto pelas colunas "Campus" e "Curso", as outras possuem apenas números.

In [ ]:
```
condicao = ufabc["Campus"] == "SA"
total_SA = ufabc[condicao].sum()["2016"]
print("Total de alunos matriculados em Santo André em 2016:", total_SA)
```

Para alterar um único elemento específico da tabela, basta saber o índice da linha e o rótulo da coluna e utilizar o comando `at[]` mencionado acima.

Por exemplo, vamos atualizar o número de matriculados no BC&T, período Noturno, em 2013 no Campus SA.

Atualmente esse valor (2732) encontra-se na linha de índice 1, mas seu valor correto é 2752.

In [ ]:
```
ufabc.at[1, "2013"] = 2752
display(ufabc)
```

Faça você mesmo

Três alunos fizeram a prova substitutiva: o aluno AB tirou nota 7.6, o aluno FU tirou nota 5.5 e o aluno GA tirou nota 0.2.

Atualize a tabela de notas com essas informações.

In [ ]:
```

```

Outra funcionalidade importante é adicionar linhas e colunas à tabela atual.

Por exemplo, vamos criar uma coluna nova que vai conter o valor médio de alunos matriculados por ano.

Com o Pandas, é bem simples criar colunas novas.

Ao acessar `ufabc[rótulo de coluna]`, se `rótulo de coluna` existe então o Pandas entende que você quer visualizar aquela coluna. Mas se o rótulo não existe, então ele entende que você quer criar uma coluna nova.

Ao mesmo tempo vamos inicializá-la com o valor da média.

Veja no exemplo a seguir que o Pandas entende que queremos fazer o cálculo linha por linha.

In [ ]:
```
ufabc["Média"] = (ufabc["2012"] + ufabc["2013"] + ufabc["2014"] + ufabc["2015"] + ufabc["2016"])/5
display(ufabc)
```

Vamos também inserir uma coluna "Total", com a soma de alunos matriculados em todos os anos anteriores:

In [ ]:
```
ufabc["Total"] = ufabc["2012"] + ufabc["2013"] + ufabc["2014"] + ufabc["2015"] + ufabc["2016"]
display(ufabc)
```

> **Extra:** É claro que se a quantidade de colunas fosse muito maior do que 5, fazer as contas anteriores do jeito que fizemos não seria viável. Mas usando os conhecimentos de laço que já vimos anteriormente, é possível fazer isso sem ter que escrever o rótulo de cada coluna. Esse código encontra-se ao fim desse notebook, caso você tenha curiosidade em saber como isso pode ser feito.

Também seria interessante inserir uma linha nova, contendo o total de alunos matriculados na UFABC em cada ano.

Lembre-se que para acessar uma linha temos o comando `loc[índice da linha]` . Se o índice existe, então o Pandas entende que você quer acessar aquela linha. Mas se o índice não existe, então ele entende que você quer criar uma linha nova.

Ao mesmo tempo vamos inicializá-la com uma lista de valores na ordem das colunas.

In [ ]:
```
ufabc.loc[12] = ["SA e SBC", "Todos", ufabc["2012"].sum(), ufabc["2013"].sum(), ufabc["2014"].sum(), ufabc["2015"].sum(), ufabc["2016"].sum(), ufabc["Média"].sum(), ufabc["Total"].su
display(ufabc)
```

> O programa anterior também pode ser reescrito para não termos que digitar um valor para cada coluna (o que é inviável quando se tem várias colunas).

Faça você mesmo

Crie uma nova coluna na tabela de notas, com a média final do aluno.

A média do aluno é calculada fazendo-se `0.8*(nota da prova 1 + nota da prova 2 + nota da prova substitutiva) + 0.2*(nota das listas)` .

In [ ]:
```

```

A penúltima funcionalidade que veremos é a remoção de colunas ou linhas.

Para isso temos a função `drop()` .

Essa função recebe uma lista que vai indicar o que deve ser removido e o comando `axis` que vai indicar de onde deve ser removido (`axis=0` se a remoção é de linhas e `axis=1` se a remoção é de colunas).

Assim, se queremos remover colunas, basta passar uma lista de rótulos das colunas que queremos remover.

Se queremos remover linhas, basta passar uma linha com índices das linhas que queremos remover.

No exemplo a seguir, vamos remover a coluna "Total", que nem faz tanto sentido assim.

Para que a remoção seja efetivada na tabela, precisamos do comando `inplace=True` também.

In [ ]:
```
ufabc.drop(["Total"], axis=1, inplace=True)
display(ufabc)
```

A seguir vamos remover todas as linhas que contêm dados de SBC.

Mas faremos isso apenas para nossa visualização, sem utilizar o comando `inplace` .

In [ ]:
```
condicao = ufabc["Campus"] == "SBC"
linhas = list(ufabc[condicao].index.values)
display(ufabc.drop(linhas, axis=0))
```

Note como a tabela original ainda contém todos os valores:

In [ ]:
```
display(ufabc)
```

A seguir vamos efetivamente remover as linhas que contêm informações sobre cursos noturnos.

In [ ]:
```
# queremos filtrar linhas que, na coluna Curso, contenham a palavra Noturno:
condicao = ufabc["Curso"].str.contains("Noturno")
linhas = list(ufabc[condicao].index.values)
ufabc.drop(linhas, axis=0, inplace=True)
display(ufabc)
```

Note que essas remoções acabaram por invalidar o conteúdo da última linha, que deveria conter um somatório dos valores presentes na tabela.

Vamos então remover a última linha e repetir o comando que a criou, lembrando que a coluna "Total" não existe mais:

In [ ]:
```
ufabc.drop([12], axis=0, inplace=True)
ufabc.loc[12] = ["SA e SBC", "Todos", ufabc["2012"].sum(), ufabc["2013"].sum(), ufabc["2014"].sum(), ufabc["2015"].sum(), ufabc["2016"].sum(), ufabc["Média"].sum()]
display(ufabc)
```

Faça você mesmo

O aluno BA desistiu da disciplina. Remova-o da tabela de notas.

In [ ]:
```

```

Faça você mesmo

No código a seguir vamos criar uma nova coluna na tabela de notas, que vai conter o conceito final de cada aluno.

Dada a média final MF de um aluno, seu conceito final será:

$$
\begin{array}{ll}
A, & \text{se } MF \geq 8.5 \\
B, & \text{se } 7.0 \geq MF < 8.5 \\
C, & \text{se } 6.0 \geq MF < 7.0 \\
D, & \text{se } 5.0 \geq MF < 6.0 \\
F, & \text{se } MF < 5.0
\end{array}
$$

Tente entender o que o comando `for` está fazendo e termine o código.

Em seguida peça para visualizar a tabela de notas.

In [ ]:
```
for i in list(notas.index.values):
    if notas.at[i, "Final"] >= 8.5:
        notas.at[i, "Conceito final"] = "A"
```

Para finalizar, pode ser útil salvar essa nova tabela que temos em mãos em um arquivo CSV também.

Para isso temos a função `to_csv()` :

In [ ]:
```
ufabc.to_csv("ufabc_alterado.csv", sep=";")
```

Vá até a pasta onde este notebook está salva (provavelmente sua pasta Downloads) e veja que o arquivo `ufabc_alterado.csv` foi criado.

Abra-o com o programa LibreOffice (por exemplo) para visualizar seu conteído.

Outras informações

Como eu disse antes, não era objetivo dessa aula vermos tudo sobre bancos de dados nem tudo sobre o Pandas.

É importante que você entenda isso: existe **muito** mais sobre esses assuntos.

Por exemplo, nessa página (https://pandas.pydata.org/pandas-docs/version/0.22/api.html) você pode ver os nomes e descrições de todas as funções oferecidas pelo Pandas.

Nós vimos as mais básicas, que cobrem várias das coisas que desejamos fazer quando temos uma tabela de dados: visualizá-la, ordená-la, adicionar ou remover valores, fazer cálculos sobre os valores existentes e filtrar conteúdos.

Extras

O código a seguir calcula a média das colunas sem termos que digitar o rótulo de cada coluna.

In [ ]:
```
# primeiro vamos pegar uma lista com os rótulos das colunas:
colunas = list(ufabc.columns.values)
# sabemos que as duas primeiras colunas possuem o campus e o curso, que não contém números para o cálculo
# sabemos também que a última coluna de média já existe (idealmente ela não existiria e a estaríamos criando agora)
# dado que colunas é uma lista, o python aceita algo do tipo colunas[x:y], significando que só queremos os itens
#    que estão armazenados entre a posição x e y
# no caso a seguir, -1 indica a última posição da lista (e não a queremos, pois ela contém a média)
ufabc["Média"] = 0
for col in colunas[2:-1]:
    ufabc["Média"] = ufabc["Média"] + ufabc[col]
# a função len() retorna o tamanho da lista
ufabc["Média"] = ufabc["Média"]/len(colunas[2:-1])
display(ufabc)
```