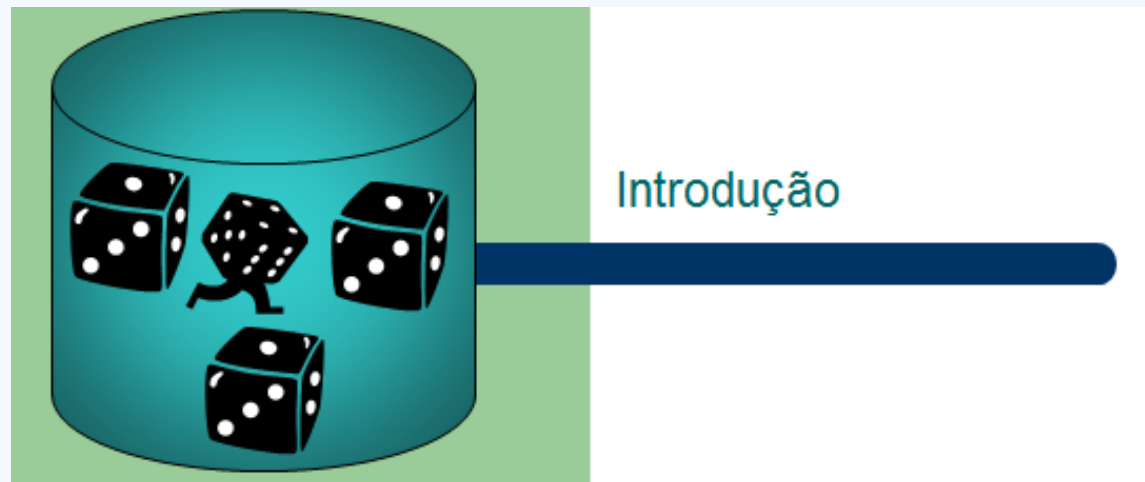


Disciplina:

PYTHON ENGENHARIA DE DADOS

Professor: Nelson Júnior





- ✓ Um sistema Gerenciador de Banco de Dados SGBD
- ✓ Primeiro SGBD comercial surgiu no final de 1960

- ✓ Sistema extremamente complexo, responsável pela persistência, organização e recuperação dos dados
- ✓ Hoje os bancos de dados podem organizar as informações: relacional e não relacional

- Definição de dados e Tipos de Dados SQL
 - Tabela, linha e coluna
 - O principal comando SQL para a definição de dados é o CREATE.
 - Esquemas
 - Tabelas (relações)
 - Domínios

- Definição de dados e Tipos de Dados SQL
 - Tabela, linha e coluna
 - O principal comando SQL para a definição de dados é o CREATE.
 - Esquemas
 - Tabelas (relações)
 - Domínios

- Tipo de dados
 - Numerico (int, smallint, float, double)
 - Cadeia de caracteres (char, varchar)
 - Bit-string
 - Boolean
 - Date e time
 - Timestamp
 - Interval

- Operações
 - Insert
 - Delete
 - Update



- ✓ O MySQL foi originalmente lançado em 1995.
- ✓ Desde 2010 à Oracle Corporation.
- ✓ Software open-source



<https://www.apachefriends.org/download.html>

XAMPP é um pacote com os principais servidores de código aberto do mercado,

Vejamos os passos que devemos seguir para realizar a conexão a um banco de dados MySQL em Python.

- ✓ Ter um SGBD instalado, configurado e em execução na máquina local. Preferencialmente.
- ✓ Instalar o conector para MySQL do Python,
`pip install mysql-connector-python`

```
'''
```

```
Created on ...
```

```
@author: nelsonjunior
```

```
'''
```

```
import mysql.connector
```

```
from banco.Professor import Professor
```

```
con = mysql.connector.connect(host='localhost', database='aula',  
user='root', password='')
```

```
class Professor:
```

```
    id = None
```

```
    nome = None
```

```
    curso = None
```

```
    turma = None
```

```
    def __init__(self):
```

```
        self.id = 0
```

```
        self.nome = ""
```

```
        self.curso = ""
```

```
        self.turma = ""
```

```
if con.is_connected():  
    db_info = con.get_server_info()  
    print( "Conectado ao servidor MySQL versão ", db_info )  
    cursor = con.cursor()
```

```
def fechaConexo():  
    if con.is_connected():  
        cursor.close()  
        con.close()  
        print( "Conexão ao MySQL foi encerrada" )
```

def incluirProfessor(professor):

```
url = "INSERT INTO professor (id, nome_professor, curso, turma) VALUES ( " + str(professor.id) + ", ' " + professor.nome + "', '" + professor.curso + "', '" + professor.turma + "'" + " )"
cursor.execute(url)
con.commit()
```

def alterarProfessor(professor):

```
url = "UPDATE professor set id = " + str(professor.id) + "," + "nome_professor = " + "'" + professor.nome + "', " + "curso = ' " + professor.curso + "', " + " turma = " + "'" + professor.turma + "'" + " Where id = " + str(professor.id)
cursor.execute(url)
con.commit()
```

```
def deleteProfessor(professor):  
    url = "delete from professor Where id = " + str(professor.id)  
    cursor.execute(url)  
    con.commit()
```

```
professor = Professor()  
professor.nome = "Vitor"  
professor.turma = "10002"  
professor.curso = "ADS 2022"  
professor.id = 10  
deleteProfessor(professor)  
fechaConexo()
```



EXERCÍCIOS



PUC Minas

Virtual