

Integrando Python com BDs relacional e NoSQL

Formação Python Developer

Juliana Mascarenhas

Tech Education Specialist DIO / Owner @Simplificandoredes e @SimplificandoProgramação

Mestre em modelagem computacional | Cientista de dados

@in/juliana-mascarenhas-ds/





<https://github.com/julianazanelatto>

Juliana Mascarenhas

Tech Education Specialist

@SimplificandoRedes

@SimplificandoProgramação

Cientista de dados

Desenvolvedora Java/Python

Me Modelagem Computacional - LNCC

Objetivo Geral

Trabalhar com a principal biblioteca ORM de integração de SGBDs com Python – SQLAlchemy. Além disso, não podemos deixar de fora os bancos de dados NoSQL. Sendo assim, utilizaremos Pymongo.

Etapa 1

Integrando Python com SQLite usando SQLAlchemy

// Integração com Python

Conhecendo a Biblioteca SQLAlchemy

// Integração com Python

SQLAlchemy



- Framework – open source
- Licença MIT – 2019
- Mapeamento Objeto Relacional



SQLAlchemy



Database	Fully tested in CI	Normal support	Best effort
Microsoft SQL Server	2017	2012+	2005+
MySQL / MariaDB	5.6, 5.7, 8.0 / 10.4, 10.5	5.6+ / 10+	5.0.2+ / 5.0.2+
Oracle	11.2, 18c	11+	8+
PostgreSQL	9.6, 10, 11, 12, 13, 14	9.6+	8+
SQLite	3.21, 3.28+	3.12+	3.7.16+

[Documentação](#)

Database	Dialect
Action Avalanche, Vector, Actian X, and Ingres	sqlalchemy-ingres
Amazon Redshift (via psycopg2)	sqlalchemy-redshift
Apache Drill	sqlalchemy-drill
Apache Druid	pydruid
Apache Hive and Presto	PyHive
Apache Solr	sqlalchemy-solr
CockroachDB	sqlalchemy-cockroachdb
CrateDB ^[1]	crate-python
EXASolution	sqlalchemy_exasol
Elasticsearch (readonly)	elasticsearch-dbapi

SQLAlchemy

- Dialectos externos

SQLAlchemy

SQLAlchemy



- Vastamento utilizado
- Framework completo
- Flexibilização do SQL
- Segurança nas instruções



SQLAlchemy

Recursos:

- ORM e CORE
- Suporte a dialetos
- Manipulação do BD por meio de Transações
- Suporte a Queries complexas via ORM
- Config: relações e relacionamentos
- Sessões, eventos ...

SQLAlchemy

Extensões:

- I/O assíncrono
- Associação com proxy
- Indexação
- APIs especiais
- ...

```
from sqlalchemy import Column, JSON, Integer
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.ext.indexable import index_property

Base = declarative_base()

class Person(Base):
    __tablename__ = 'person'

    id = Column(Integer, primary_key=True)
    data = Column(JSON)

    name = index_property('data', 'name')
```

[Documentação](#)

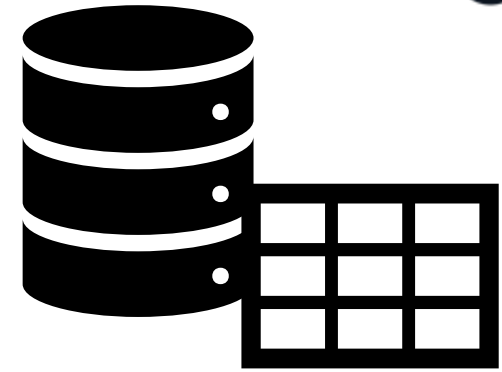
ORM

- ORM - Object Relational Mapping
- Objeto -> Modelo Relacional
- Mais fácil para o programador

CRUD

QUERY

CONEXÃO



ORM

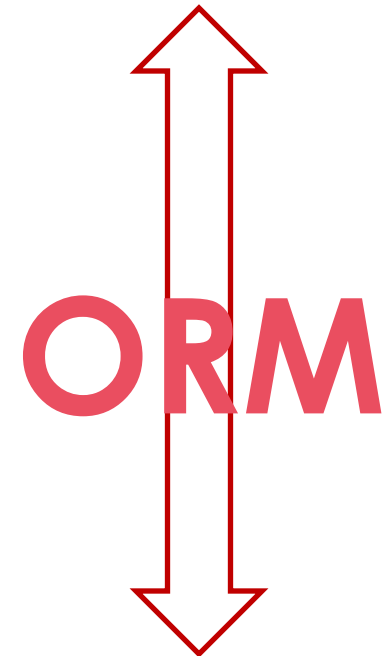
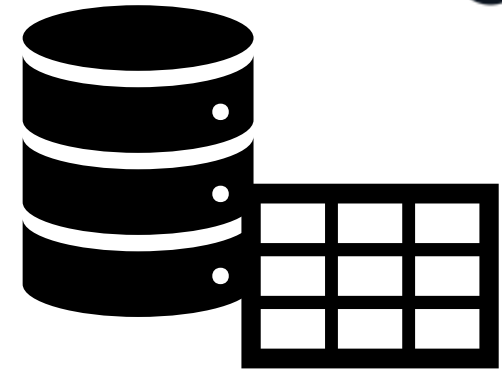
{OOP}

ORM

Vantagens

- Menos código
- Melhor manutenção
- Utilização de conectores
- Indicado para CRUDs

SQLAlchemy



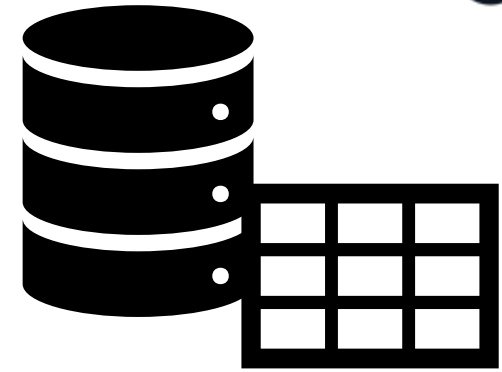
{ OOP }

Entidade

```
>>> from sqlalchemy import Column, Integer, String
>>> class User(Base):
...     __tablename__ = 'users'
...
...     id = Column(Integer, primary_key=True)
...     name = Column(String)
...     fullname = Column(String)
...     nickname = Column(String)
...
...     def __repr__(self):
...         return "<User(name='%s', fullname='%s', nickname='%s')>" % (
...             self.name, self.fullname, self.nickname)
```

ORM

Se pergunte como está
seu modelo de dados



Desvantagens

- Complexidade X ORM
- Dependência do ORM
- Depende do projeto
- Retorno das consultas sem necessidade de programar na "mão"

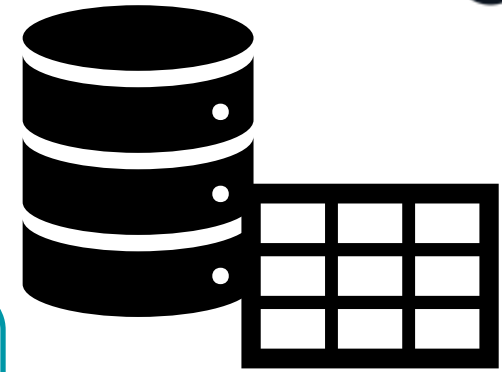
SQLAlchemy

ORM

{ OOP }

ORM

SQLAlchemy



Nem sempre será a query mais otimizada

- Perda de performance
- Deixa de estudar SQL e perde a eficiência na construção
- Número de instâncias x velocidade

ORM

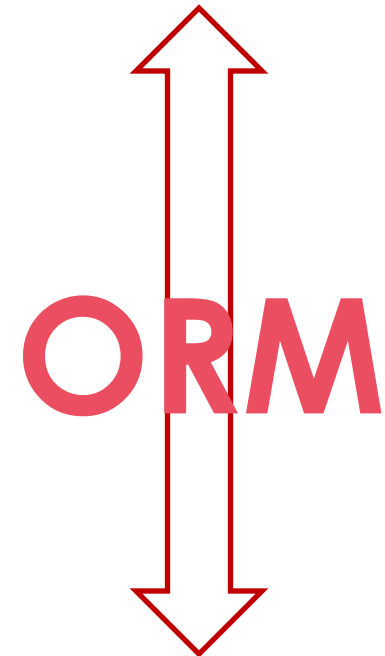
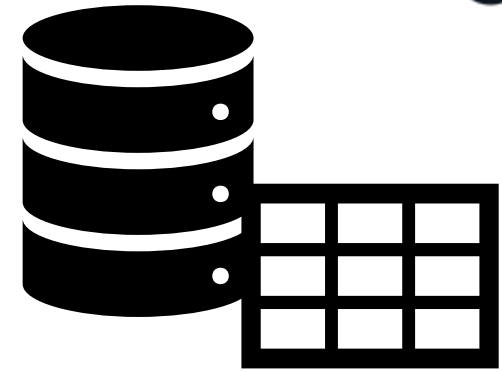
{OOP}

ORM

Por que usar?

- Troca de SGBD mais facilitada
- Modelo MVC
- Diminuição do DRY
- Evita problemas de segurança

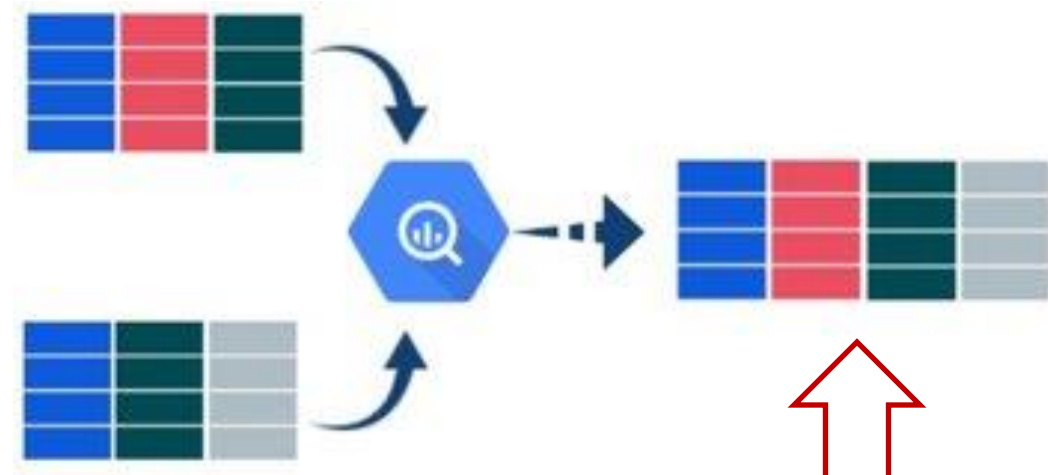
SQLAlchemy



{OOP}

ORM & SQL

Uso de views



- Melhor dos dois mundos
- Qual a melhor ferramenta para o seu problema?

ORM

{OOP}

Dúvidas?

- > Fórum/Artigos
- > Comunidade Online (Discord)



Para saber mais

Python e SQLite

- <https://www.sqlalchemy.org/>
- <https://docs.sqlalchemy.org/en/14/tutorial/index.html>
- <https://docs.sqlalchemy.org/en/14/orm/quickstart.html>
- <https://docs.sqlalchemy.org/en/14/orm/tutorial.html>



Para saber mais

Dialetos

- <https://docs.sqlalchemy.org/en/14/dialects/index.html>
- SQLite - <https://docs.sqlalchemy.org/en/14/dialects/sqlite.html>
- MySQL - <https://docs.sqlalchemy.org/en/14/dialects/mysql.html>

Subqueries e demais recursos

- https://docs.sqlalchemy.org/en/14/tutorial/data_select.html#tutorial-select
[data](#)



Para saber mais

Modelo ORM

- <https://docs.sqlalchemy.org/en/14/orm/index.html>
- <https://docs.sqlalchemy.org/en/14/orm/quickstart.html>

Modelo Core

- <https://docs.sqlalchemy.org/en/14/core/index.html>
- <https://docs.sqlalchemy.org/en/14/core/tutorial.html>

