

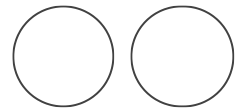


Ferramentas ORM, SQL ANSI e Engenharia de Bancos de Dados



Igor Felix

Computer Scientist - Mobile Developer - .Net(Fullstack) - .Net(MAUI),
Linux Developer and Python(Django) - NodeJS(Express) - BAT/BASH -...



10 de maio de 2025

1. Introdução à Engenharia de Bancos de Dados

A Engenharia de Bancos de Dados é uma disciplina essencial da Ciência da Computação voltada para o projeto, implementação, manutenção e otimização de bancos de dados. Desde os primeiros SGBDs como o DBASE, que popularizou o uso de bancos de dados em sistemas pessoais nos anos 1980, até o MySQL, um dos SGBDs relacionais mais utilizados no mundo atual, houve uma evolução significativa na forma como dados são armazenados, consultados e gerenciados.

2. Diferenças entre Providers: ODBC vs Connectors

ODBC (Open Database Connectivity) é uma API padronizada que permite a aplicações acessarem bancos de dados independentemente do SGBD. Já os connectors são APIs específicas para linguagens e SGBDs particulares (como o MySQL Connector para Python). Enquanto o ODBC oferece portabilidade, os connectors proporcionam maior performance e acesso a funcionalidades específicas do banco de dados.

3. Início do ORM com o Hibernate

O ORM (Object-Relational Mapping) surgiu como solução para o "impedance mismatch" entre modelos orientados a objetos e bancos de dados relacionais. O Hibernate, desenvolvido originalmente por Gavin King e adotado pela Oracle, foi uma das primeiras ferramentas ORM de destaque, proporcionando uma abstração robusta entre objetos Java e tabelas relacionais.

4. Importância da Engenharia de Bancos de Dados

A engenharia de bancos de dados é fundamental para garantir integridade, consistência e eficiência no acesso aos dados. Um banco bem projetado reduz redundâncias, melhora a performance e facilita a manutenção e expansão de sistemas.

5. SGBDs e Cientistas de Dados

Para cientistas de dados, um SGBD é vital para organizar, consultar e analisar grandes volumes de dados. Ferramentas como PostgreSQL e SQLite são amplamente usadas devido à sua robustez e compatibilidade com ferramentas de análise de dados como Python e R.

6. Teoria do ORM e Compatibilidade entre SGBDs

O ORM baseia-se na representação de entidades como objetos de classes e seus atributos como colunas. A compatibilidade entre diferentes SGBDs (como PostgreSQL, MySQL, SQLite e Oracle) é um desafio enfrentado por ferramentas ORM modernas, que precisam abstrair diferenças de dialetos SQL e tipos de dados.

7. Criação de Bases em SQLite com Python e CSV

Python oferece bibliotecas como sqlite3 e pandas que facilitam a criação de bancos de dados a partir de arquivos CSV. Exemplo:

```
import pandas as pd
import sqlite3

conn = sqlite3.connect('banco.db')
```

```
df = pd.read_csv('dados.csv')
df.to_sql('tabela', conn, if_exists='replace', index=False)
```

Consultas SQL ANSI com filtros:

```
SELECT * FROM tabela WHERE data BETWEEN '2024-01-01' AND '2024-12-31'
ORDER BY resultado DESC;
SELECT tipo, COUNT(*) FROM tabela GROUP BY tipo ORDER BY COUNT(*) ASC;
UPDATE tabela SET status='confirmado' WHERE id=10;
```

Armazenar HTML como blob:

```
html_result = '<html><body>Relatório</body></html>'
cursor = conn.cursor()
cursor.execute("INSERT INTO relatorios (conteudo) VALUES (?)",
[html_result.encode('utf-8')])
conn.commit()
```

8. Modelo MVVM para ORM

O modelo MVVM (Model-View-ViewModel) é uma arquitetura comumente usada em aplicações com ORM, onde o Model representa as entidades do banco, o ViewModel atua como intermediário e o View é responsável pela interface. Com ORM, os Models são usados para gerar e atualizar as tabelas automaticamente.

9. Operações Triviais do SQL ANSI

O SQL ANSI define operações básicas como SELECT, INSERT, UPDATE, DELETE, CREATE e DROP. Essas operações garantem interoperabilidade entre diferentes sistemas de bancos de dados e são fundamentais no uso de qualquer ferramenta ORM.

10. Criptografia em ORM com BASE64 e RSA

Embora BASE64 não seja uma técnica de criptografia, mas sim de codificação, é usada para transportar dados binários como strings. A criptografia real pode ser implementada com RSA para proteger dados sensíveis em aplicações ORM. Bibliotecas como cryptography ou PyCryptodome permitem integração com ORM como SQLAlchemy.

11. OntoDBs e Dados Baseados em Ontologias

OntoDBs são bancos de dados baseados em ontologias que integram conceitos semânticos diretamente na estrutura do banco. Exemplos incluem Virtuoso, AllegroGraph e OntoDB. Proposições modernas incluem o uso de coleções de objetos e tuplas para representar entidades, favorecendo a flexibilidade em aplicações de IA e Web Semântica.

12. Scikit-learn para Classificação e Agrupamento em Pesquisas Online

Com scikit-learn, é possível classificar informações da web com base em categorias de globalização, utilizando técnicas como KMeans e RandomForest para distinguir conteúdos. Exemplo:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.cluster import KMeans

textos = ["produto nacional", "produto importado", "tecnologia local",
"marca global"]
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(textos)
model = KMeans(n_clusters=2)
model.fit(X)
print(model.labels_)
```

13. OpenAI e Geração de Modelos ER com Prompts

Com modelos de linguagem como os da OpenAI, é possível gerar modelos ER com base em prompts textuais. Exemplo com prompt Python:

```
pip install openai

import openai
```

```
# Substitua 'sua-chave-api' pela sua chave de API da OpenAI
openai.api_key = 'sua-chave-api'

response = openai.Completion.create(
    engine="text-davinci-003", # Escolha o modelo desejado
    prompt="Escreva um poema sobre a natureza.",
    max_tokens=100 # Número máximo de tokens na resposta
)

print(response.choices[0].text.strip())
```

Esse retorno pode ser salvo como CSV ou TXT com a descrição de entidades e relações.

14. Edgar F. Codd e a Normalização

Edgar Frank Codd foi o matemático e cientista da computação que propôs o modelo relacional e a teoria da normalização de dados, alicerces da Engenharia de Bancos de Dados moderna. Seus diagramas de Entidade-Relacionamento (ER) permitiram abstrair a estrutura de dados de forma intuitiva e eficaz.

15. 🧠 Python Database X-AI ORM

Com ajuda do repositório de arquivos para programadores de computador, existe uma vasta gama de aplicativos de software e monolíticos disponíveis na internet.

O Python Database X-AI ORM um projeto Open-Source, baseado na licença Creative Commons Zero, oferece uma base para o estudo do ORM e o MVVM, contando com o apoio da AI para Engenheiros de Bancos de Dados e Cientistas de Dados.

Sobre o Python Database X-AI ORM:

É uma ferramenta, desenvolvida por #asytrick em Python, para mapeamento objeto-relacional (ORM) para SQLite com funcionalidades voltadas à simplicidade, produtividade e inteligência artificial. O projeto visa acelerar o desenvolvimento de bancos de dados relacionais com:

- Instalação de modelos ER via arquivos dicionário.

- Execução de operações triviais com apenas um comando.
- Geração e atualização de modelos compatíveis com auxílio de prompts da OpenAI.
- Apoio à evolução de esquemas e integração futura com bibliotecas de ciência de dados e aprendizado de máquina.

O projeto está disponível na World Wild Web e também no git:

```
git clone https://github.com/ssmool/PY_DBXIA ORM.git
cd PY_DBXIA ORM
python install.py # (if applicable)
```

Conclusão

A evolução da Engenharia de Bancos de Dados e o surgimento de ferramentas ORM proporcionaram agilidade e abstração na construção de sistemas complexos. O entendimento de tecnologias como SQL ANSI, criptografia, modelos arquiteturais e ontologias semânticas permite o desenvolvimento de soluções robustas, seguras e escaláveis.