

Bases de Dados

T19 - Desenvolvimento Web e Bases de Dados

Prof. Daniel Faria

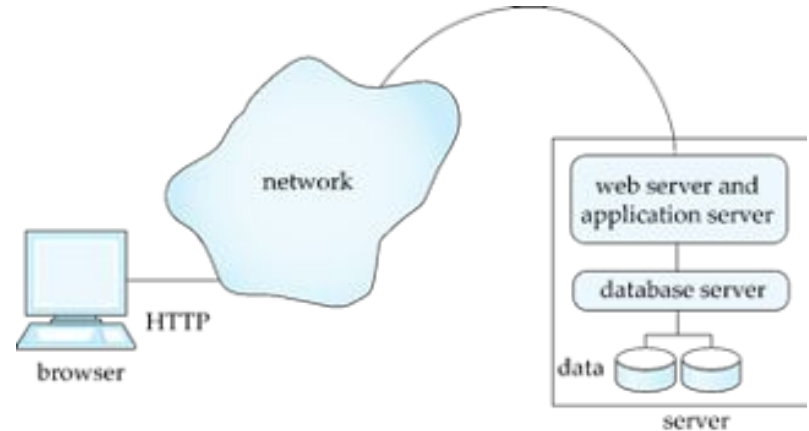
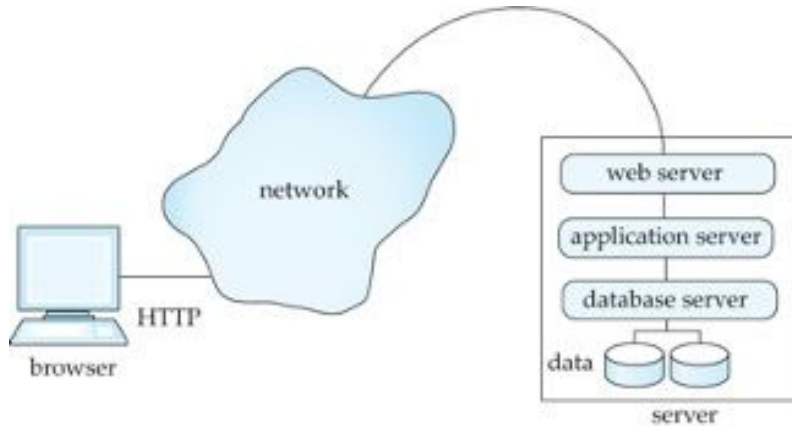
Prof. Flávio Martins

Sumário

- Recapitulação Breve
- Desenvolvimento Web e Bases de Dados
- bank-app em Flask

Recapitulação Breve

Arquitetura de Aplicações Web



Psycopg Prepared Statement: Exemplo

Utilizando os objectos do Psycopg com context managers (i.e. com `with`) garante o fecho automático das conexões e a libertação dos recursos no final do bloco (no `psycopg3`).

- `connect()` retorna uma `Connection`
 - `conn.cursor()` retorna `Cursor`
 - `conn.commit()` e `conn.rollback()`
- Executar SQL
 - `cur.execute()`
 - `cur.executemany()`
- Ir buscar conjuntos de tuplos
 - `cur.fetchone()`
 - `cur.fetchmany()`

```
1  # Note: the module name is psycopg, not psycopg3
2  import psycopg
3
4  # Connect to an existing database
5  with psycopg.connect("dbname=test user=postgres") as conn:
6
7      # Open a cursor to perform database operations
8      with conn.cursor() as cur:
9
10         # Pass data to fill a query placeholders and let Psycopg perform
11         # the correct conversion (no SQL injections!)
12         cur.execute(
13             "INSERT INTO test (num, data) VALUES (%s, %s)",
14             (100, "abcdef"))
15
16         # Query the database and obtain data as Python objects.
17         cur.execute("SELECT * FROM test")
18         cur.fetchone()
19         # will return (1, 100, "abcdef")
20
21         # You can use `cur.fetchmany()`, `cur.fetchall()` to return a list
22         # of several records, or even iterate on the cursor
23         for record in cur:
24             print(record)
25
26         # Make the changes to the database persistent
27         conn.commit()
```

Psycopg

com context manager

Utilizando os objectos do Psycopg com context managers (i.e. com `with`) garante o fecho automático das conexões e a libertação dos recursos no final do bloco (no `psycopg3`).

- É inserido um record na tabela
- Ao sair do contexto de `with .. as conn`
 - `commit()`
 - `close()`

```
with psycopg.connect() as conn:

    cur = conn.cursor()

    cur.execute("SELECT count(*) FROM my_table")
    # This function call executes:
    # - BEGIN
    # - SELECT count(*) FROM my_table
    # So now a transaction has started.

    cur.execute("INSERT INTO data VALUES (%s)", ("Hello",))
    # This statement is executed inside the transaction

# No exception at the end of the block:
# COMMIT is executed.
```

Desenvolvimento Web Bases de Dados

Server-Side Web Frameworks

- Python
 - Django
 - Flask
- Java
 - Spring Boot
 - Play
- Express for JavaScript
- Ruby on Rails
- Laravel for PHP

Django

- O Django segue a filosofia "batteries included" e fornece implementações próprias de quase tudo o que pode.
- Sites populares: Instagram, Pinterest (-2015)



Flask

- Flask é um microframework para Python e fornece apenas o estritamente necessário para poder criar sites:
 - Servidor de desenvolvimento e depurador
 - Suporte para templates Jinja2.
- Patreon, Pinterest (2016-), Reddit (migração de Pylons)



Create, Read, Update and Delete (CRUD)

O acrónimo CRUD refere-se às principais operações implementadas por bases de dados. Cada letra do acrónimo corresponde a um statement de Structured Query Language (SQL).

| CRUD | SQL |
|--------|--------|
| Create | INSERT |
| Read | SELECT |
| Update | UPDATE |
| Delete | DELETE |

Create, Read, Update and Delete (CRUD)

O acrónimo CRUD é também usado em APIs RESTful. Cada letra do acrónimo pode corresponder a um método de Hypertext Transfer Protocol (HTTP)

| CRUD | HTTP |
|--------|--|
| Create | POST, PUT se tivermos <code>id</code> ou <code>uuid</code> |
| Read | GET |
| Update | PUT para substituir, PATCH para modificar |
| Delete | DELETE |

Referências

- <https://developer.mozilla.org/en-US/docs/Web/HTML>
- <https://developer.mozilla.org/en-US/docs/Web/HTTP>
- <https://flask.palletsprojects.com/en/2.3.x/>