



Banco de Dados II

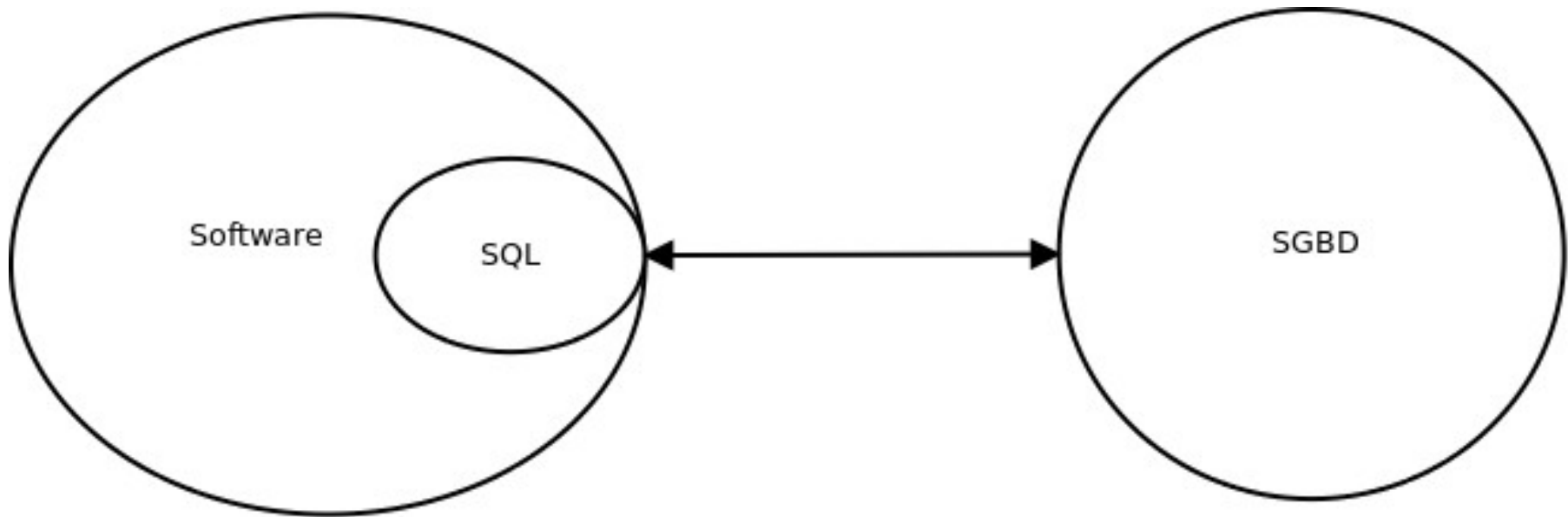
Prof. Vinícius Alves Hax



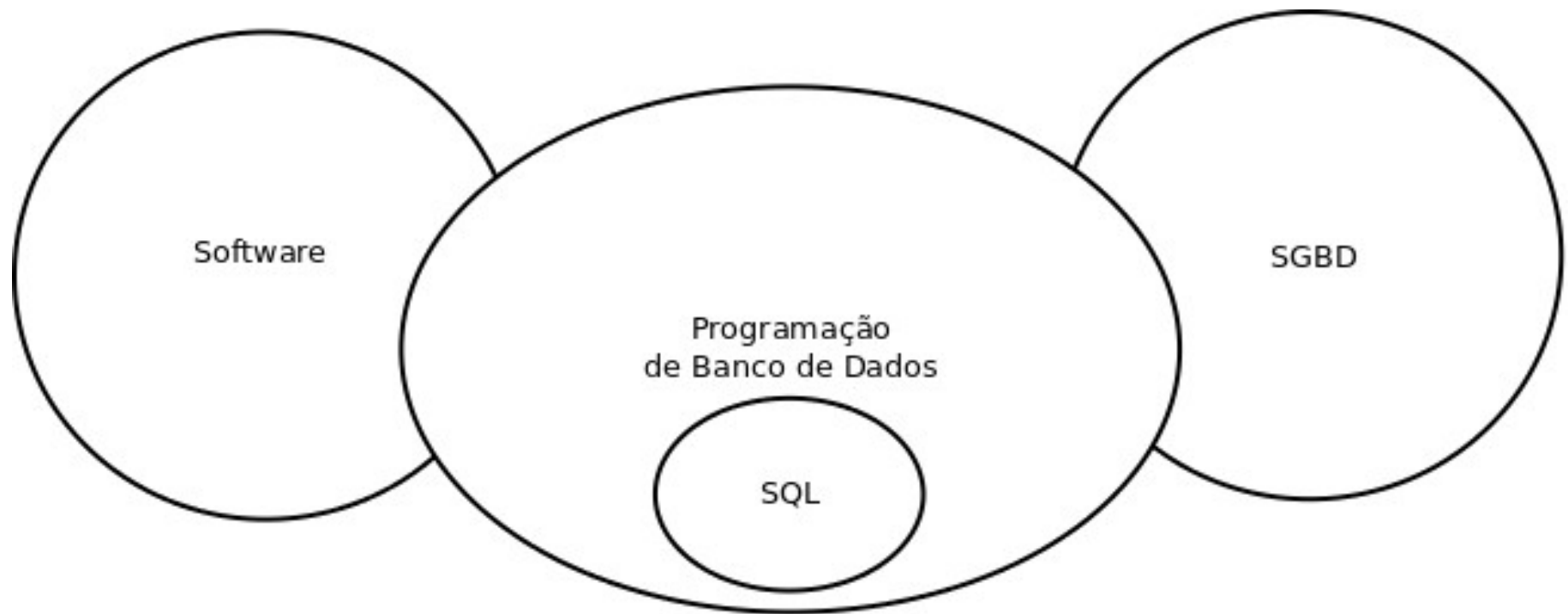
Na aula de hoje

- Programação dentro do banco de dados

Até onde vimos:



Na verdade é mais ...





SQL e suas limitações

SQL é muito interessante e resolve 80% dos problemas porém tem algumas limitações.

Por exemplo não temos, dentro do SQL, if e for

Para esses 20% de situações temos a programação em banco de dados: como se fosse uma linguagem de programação, com regras próprias, dentro do SGBD



Motivos para usar programação em banco de dados

- Operações complexas
- Processamento em lote
- Customização
- Eficiência
- Múltiplas interfaces



Operações complexas

- SQL é muito interessante mas pode não ser difícil fazer algumas operações mais específicas
 - Exemplo: Fazer update em um registro dependendo do valor de alguma coluna

Processamento em lote

- O que é processamento em lote? Operações que não precisam ser feitas no mesmo momento e podem ser agrupadas por eficiência.
 - Ex: fechamento de fatura de cartão, geração de folha de pagamento
- Essas operações podem ser agendadas para horários de menor uso do SGBD, mas nesse caso não existe um programa interativo. É preciso programar “dentro” do SGBD



Outras:

- Customização: Permitir à ferramentas externas ir além do SQL
- Eficiência: para uma quantidade muito grande de registros, pode ser necessário otimizar a consulta
- Padronização: se eu tiver múltiplos softwares usando a mesma base de dados eu posso fazer a programação em um único lugar

PL/SQL e PL/pgSQL

- A programação para banco de dados é feita usando uma linguagem própria. A principal é o PL/SQL (Procedural Language/Structured Query Language) que executa em bancos de dados Oracle
 - No PostgreSQL temos um padrão ligeiramente diferente chamado PL/pgSQL

Estrutura básica

-- PL/SQL

DECLARE

-- declarações de
variáveis

BEGIN

-- corpo do bloco

END

-- PL/pgSQL

DO \$\$


DECLARE

-- declarações

BEGIN

-- corpo do bloco

END \$\$



Como o PostgreSQL é um banco de dados gratuito, ao contrário do Oracle, a partir deste momento passaremos a usar a sintaxe específica do Postgres. Se for usar outro banco de dados podem ser necessárias adaptações



Funções anônimas

Se quisermos utilizar um bloco uma única vez, podemos declará-lo como uma função anônima, ou seja, sem nome.

A função não precisa ter nome pois não iremos chamá-la novamente mais tarde



Funções anônimas em PL/pgSQL

```
[ <<rótulo>> ]
```

```
[ declare
```

```
    declarações ]
```

```
begin
```

```
    comandos;
```

```
...
```

```
end [ rótulo ];
```



Nossa primeira função anônima

do \$\$

begin

-- Mostra a mensagem 'Olá mundo'

raise notice 'Olá mundo';

end \$\$;


Integrando PL/pgSQL e SQL

```
do $$  
declare  
    contador integer := 0;  
begin  
    -- conta o número de registros  
    select count(*) as contador  
    into contador  
    from teacher;  
  
    -- display a message  
    raise notice 'Número é %', contador;  
end $$;
```




Exercício em aula

Como eu poderia usar PL/pgSQL para mostrar o id do professor cujo atributo “name” é “Luke”?



```
do $$  
declare  
    codigo integer := 0;  
begin  
    -- conta o número de registros  
    select teacher.id  
    into codigo  
    from teacher  
    where name = 'Luke';  
  
    -- display a message  
    raise notice 'Número é %', codigo;  
end $$;
```



Referências

- <https://www.devmedia.com.br/stored-procedures-no-postgresql-pl-pgsql-e-sobrecarga-de-funcao/6906>
- <https://www.profissionaloracle.com.br/2023/07/31/diferencas-praticas-entre-postgresql-pl-pgsql-e-oracle-pl-sql/>
- <https://www.postgresqltutorial.com/postgresql-plpgsql/plpgsql-block-structure/>



Referências

- <https://pt.wikipedia.org/wiki/SQL>
- https://www.tutorialspoint.com/postgresql/postgresql_delete_query.htm