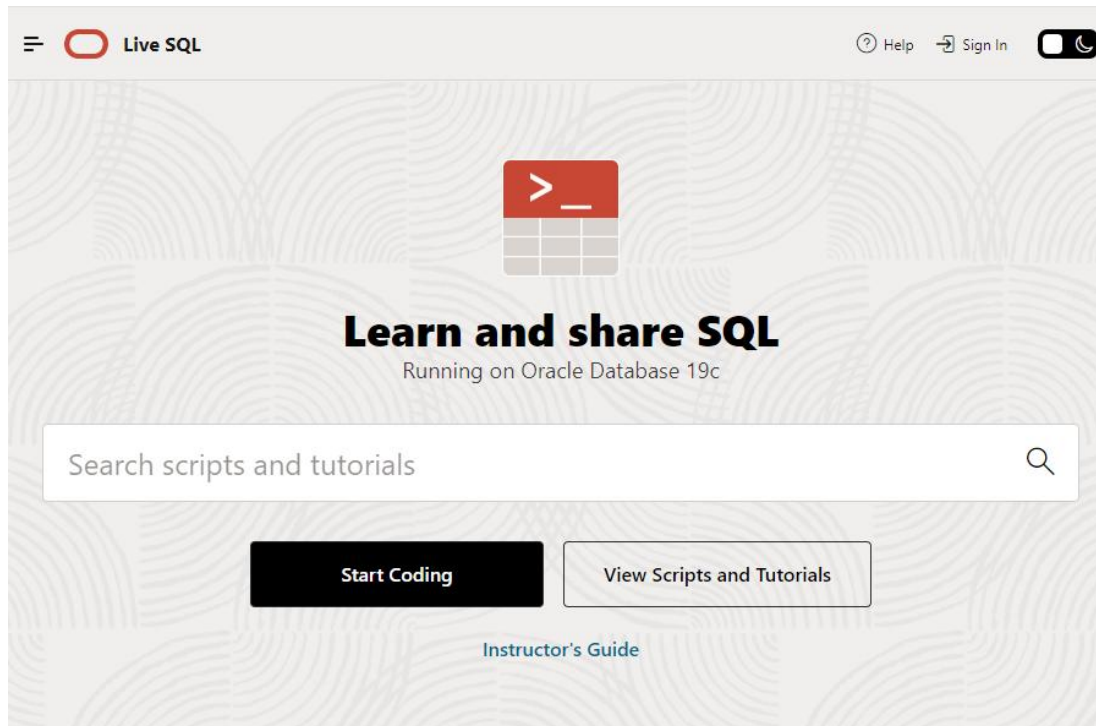


# Laboratório de Banco de Dados Avançado

# Revisão de conteúdo



- Fazer login no Live SQL da Oracle
- <https://livesql.oracle.com/>

# case when

```
1 ✓ CREATE TABLE tbProduto (  
2     idProduto NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,  
3     produto VARCHAR2(50),  
4     valor NUMBER(10,2)  
5 );  
6  
7 -- Inserir os valores  
8 INSERT INTO tbProduto VALUES (default, 'Smartphone', 899.99);  
9 INSERT INTO tbProduto VALUES (default, 'Fone de Ouvido', 299.99);  
10 INSERT INTO tbProduto VALUES (default, 'Televisão', 1499.99);  
11 INSERT INTO tbProduto VALUES (default, 'Livro de Programação', 59.90);  
12 INSERT INTO tbProduto VALUES (default, 'Romance Histórico', 34.90);  
13  
14 select * from tbProduto;
```

IDPRODUTO	PRODUTO	VALOR
1	Smartphone	899.99
2	Fone de Ouvido	299.99
3	Televisão	1499.99
4	Livro de Programação	59.9
5	Romance Histórico	34.9

# case when

```
select produto,  
       valor,  
       case  
         when (valor > 0 and valor < 50) then 'Valor Baixo' ← Condição 1  
         when (valor >= 50 and valor < 100) then 'Intermediário' ← Condição 2  
         else 'Valor Alto'  
       end classificacao  
from produto;
```

## Exemplo de Saída

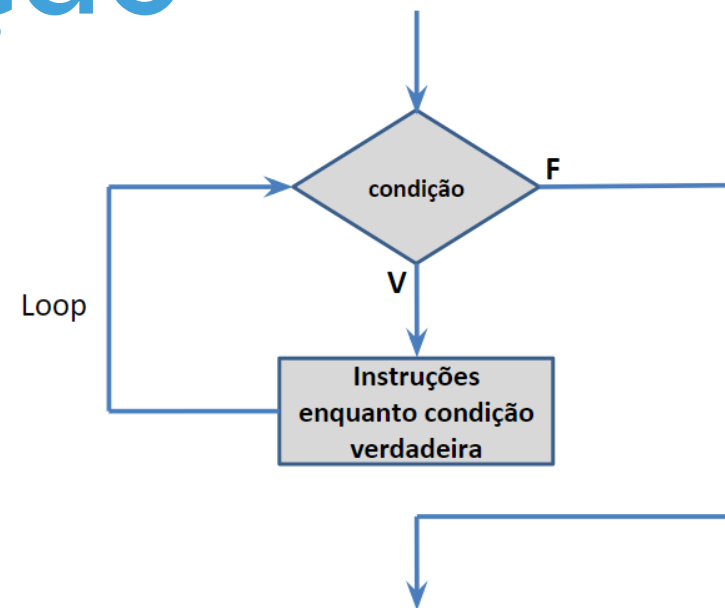
PRODUTO	VALOR	CLASSIFICACAO
Smartphone	899.99	Valor Alto
Fone de Ouvido	299.99	Valor Alto
Televisão	1499.99	Valor Alto
Livro de Programação	59.9	Intermediário
Romance Histórico	34.9	Valor Baixo

# Exercícios

1. Adicione a coluna estoqueProduto na tabela tbProduto.
2. Atualize cada uma das linhas com valores para a coluna estoqueProduto.
3. Utilizando case when, crie uma consulta que exiba as colunas, produto, valor e situacao. Com estoque abaixo de 10 exibir ESTOQUE MUITO BAIX, Com estoque igual a 0 exibir SEM ESTOQUE, caso contrário exibir DISPONÍVEL.

# Laços de Repetição

while, loop e for



# while

```
1  -- while
2  declare
3      base number := 2;
4      expoente number := 5;
5      contador number := 1;
6      potencia number :=1;
7  begin
8      while contador <= expoente loop
9          potencia := potencia * base;
10         contador := contador +1;
11     end loop;
12
13     dbms_output.put_line(potencia);
14 end;
```

Statement processed.

32

# loop

```
1  -- loop
2  DECLARE
3      contador number := 1;
4      numero number := 5;
5  BEGIN
6      LOOP
7          EXIT WHEN contador > 10;
8          DBMS_OUTPUT.PUT_LINE(numero || ' * ' || contador || ' = ' || (numero*contador));
9          contador := contador + 1;
10     END LOOP;
11 END;
```



# for

```
1  -- for
2  DECLARE
3      fatorial number :=1;
4      numero number := 6;
5  BEGIN
6      FOR i IN 1..numero LOOP
7          fatorial := fatorial * i;
8      END LOOP;
9
10     dbms_output.put_line(fatorial);
11 END;
```

# Array

- Um vetor em PL/SQL pode ser representado como:
  - **tabelas associativas** – utilizada para manipular os dados em memória;
  - **varrays** – utilizada para pequenas coleções, ordenada e com tamanho fixo;
  - **nested tables** – utilizada para guardar listas dentro de tabelas do banco sem tamanho fixo.

# Exemplo com tabela associativa

```
1 v DECLARE
2     TYPE t_numeros IS TABLE OF NUMBER INDEX BY PLS_INTEGER;
3     v_numeros t_numeros;
4 v BEGIN
5     v_numeros(1) := 5;
6     v_numeros(2) := 10;
7
8     DBMS_OUTPUT.PUT_LINE('Número: ' || v_numeros(1));
9     DBMS_OUTPUT.PUT_LINE('Número: ' || v_numeros(2));
10 END;
```

# Exemplos de Departamentos

- /\*
- Tecnologia
- Administrativo
- Financeiro
- Recursos Humanos
- Comercial
- Marketing
- Produção
- Jurídico
- Logística
- Atendimento
- Pesquisa e Desenvolvimento
- \*/

# Tabela

```
create table tbDepartamento(  
    idDepartamento number primary key,  
    departamento varchar2(40)  
);
```

# Exemplo de Inserção com uso de Laço e VArray

```
declare
    type t_departamento is varray(5) of varchar2(50);
    v_departamento t_departamento := t_departamento(
        'Tecnologia',
        'Administrativo',
        'Financeiro',
        'Recursos Humanos',
        'Comercial'
    );
begin
    for i in 1..5 loop
        insert into tbDepartamento values(i,v_departamento(i));
    end loop;
end;
```

# Exemplo com nested table

```
1 CREATE OR REPLACE TYPE numerosTable IS TABLE OF NUMBER;
2
3 DECLARE
4     vNumeros numerosTable := numerosTable(1, 2, 3, 4);
5 BEGIN
6     FOR i IN 1 .. vNumeros.COUNT LOOP
7         DBMS_OUTPUT.PUT_LINE('Posição ' || i || ' = ' || vNumeros(i));
8     END LOOP;
9
10    -- Excluir um valor
11    vNumeros.DELETE(2);
12
13    DBMS_OUTPUT.PUT_LINE('Após a exclusão ');
14
15    DBMS_OUTPUT.PUT_LINE('Total de números: ' || vNumeros.COUNT);
16
17    FOR i IN vNumeros.FIRST .. vNumeros.LAST LOOP
18        IF vNumeros.EXISTS(i) THEN
19            DBMS_OUTPUT.PUT_LINE('Posição ' || i || ' = ' || vNumeros(i));
20        END IF;
21    END LOOP;
22
23    -- Inserir valor
24    vNumeros.EXTEND;
25    vNumeros(vNumeros.LAST) := 50;
26
27    DBMS_OUTPUT.PUT_LINE('Após a inserção ');
28    FOR i IN 1 .. vNumeros.COUNT LOOP
29        IF vNumeros.EXISTS(i) THEN
30            DBMS_OUTPUT.PUT_LINE('Posição ' || i || ' = ' || vNumeros(i));
31        END IF;
32    END LOOP;
33 END;
```

# Inserindo valores em uma tabela utilizando bloco PL/SQL

```
create table DEPARTMENTS (  
  deptno      number,  
  name        varchar2(50) not null,  
  location    varchar2(50),  
  constraint pk_departments primary key (deptno)  
);
```

```
declare  
  numero number := 1;  
  nome varchar2(50) := 'RH';  
  localizacao varchar2(50) := 'São Paulo';  
begin  
  insert into DEPARTMENTS VALUES (numero,nome,localizacao);  
  DBMS_OUTPUT.PUT_LINE('Departamento ' || nome || ' incluído com sucesso');  
end;
```



# Alterando valores em uma tabela utilizando bloco PL/SQL

```
-- exemplo de pl/sql com update
declare
    var_numero number := 1;
    var_nome varchar2(50) := 'RH';
    var_localizacao varchar2(50) := 'São Paulo';
begin
    update departments set
        name = var_nome,
        location = var_localizacao
        where deptno = var_numero;

    dbms_output.put_line('Departamento alterado com sucesso');
end;
```

# Excluindo linhas em uma tabela utilizando bloco PL/SQL

```
-- exemplo de pl/sql com delete
declare
    var_numero number := 1;
begin
    delete from departments
        where deptno = var_numero;

    dbms_output.put_line('Departamento excluído com sucesso');
end;
```

# Contando linhas utilizando bloco PL/SQL

```
declare
    var_nome varchar2(50) := 'RH';
    var_total number;
begin
    select count(*) into var_total from departments
    where name = var_nome;
    dbms_output.put_line('Total de departamentos: ' || var_total);
end;
```

# Cursor

- É um comando do PL/SQL que permite a construção de uma estrutura de repetição, onde, pode-se varrer uma tabela, linha por linha, coluna por coluna através da utilização deste comando, assim, podemos manipular todos os dados de uma determinada tabela.
- Os retornos dos dados apresentados pelo cursor provem de comandos *selects* feitos em determinadas tabelas e que trazem dados de linhas e colunas específicas.

# Tipos de cursores

- Explícitos
- Implícitos

# Cursor Explícito

- É o que definimos nos programas;
- Quando usado o cursor explícito, o desenvolvedor, no caso quem está escrevendo a aplicação, além de declará-lo deve inserir comandos que especifiquem a sua inicialização e finalização, bem como a manipulação dos dados;

# Cursor Implícito

- Quando não existir um cursor explícito associado para o comando SQL, o Oracle o cria implicitamente.
- Um exemplo disto é quando temos um comando *update* ou *delete*, dentro de uma aplicação. Embora, não seja visível explicitamente, o Oracle cria um cursor para executar tal comando

# Exemplo de bloco PL/SQL com uso de Cursor

```
DECLARE
  CURSOR dept_cursor IS
    SELECT deptno, name, location
    FROM DEPARTMENTS;

  v_deptno DEPARTMENTS.deptno%TYPE;
  v_name DEPARTMENTS.name%TYPE;
  v_location DEPARTMENTS.location%TYPE;
BEGIN

  OPEN dept_cursor;

  LOOP
    FETCH dept_cursor INTO v_deptno, v_name, v_location;
    EXIT WHEN dept_cursor%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE('Deptno: ' || v_deptno || ', Name: ' || v_name || ', Location: ' || v_location);
  END LOOP;
  CLOSE dept_cursor;
END;
```



# Exercícios

1. Crie um bloco PL/SQL que faça o cadastro de uma categoria caso um idCategoria especificado não exista. Caso contrário exibir uma mensagem que o idCategoria já existe.
2. Utilizando cursor crie um bloco PL/SQL que exiba todos os dados da tabela tbProduto.

# Resumindo os exercícios...

- No mínimo
- Fazer os exercícios do slide 5 (case when)
- Fazer os exemplos dos slides 13 e 14 (laço e array)
- Fazer os exercícios do slide 25

# Contato

E-mail: [allan@cruzeirosul.edu.br](mailto:allan@cruzeirosul.edu.br)



# Universidade **Cruzeiro do Sul**

[www.cruzeirodosul.edu.br](http://www.cruzeirodosul.edu.br)