

GERENCIAMENTO DE DADOS E INFORMAÇÃO

CONSULTAS SQL

Rafael do Nascimento Moura (rnm4@cin.ufpe.br)

Rodrigo Rocha Moura (rrm2@cin.ufpe.br)

Recife, 18 de Maio de 2022

CONSULTA USANDO GROUP BY E HAVING:

```
create procedure Proc3 (parametro number) is
  cursor Cursor1 is
    select tipo_1, avg(P_ataque) as ataque
    from Pokemon
    group by tipo_1
    having avg(P_ataque) > parametro;
begin
  for consulta1 in Cursor1 loop
    dbms_output.put_line('Tipo:||consulta1.tipo_1||'
      Ataque:||consulta1.ataque);
  end loop;
end Proc3;
```

CONSULTA USANDO JUNÇÃO INTERNA:

```
create or replace function Func
(parametro number) return number is
  qtd_pokemon number := 0;
begin
  select count(*) into qtd_pokemon
  from Treinador T inner join Pokemon P on
  T.tcn = P.capturado_por
  where T.ranking = parametro;
  return qtd_pokemon;
end;
```

CONSULTA USANDO JUNÇÃO EXTERNA:

```
create or replace procedure Proc (data_in date) is
  cursor Cursor1 is
    select D.duelo_data, D.codigo_Insignia,
           D.tcn_vencedor
    from Duelo D left join Insignia I
    on D.codigo_Insignia = I.codigo_da_Insignia
    where D.duelo_data > data_in;
begin
  for x in Cursor1 loop
    if x.codigo_Insignia is not null then
      dbms_output.put_line('Data:
'||x.duelo_data||'; Insignia:
'||x.codigo_Insignia||'; Vencedor:
'||x.tcn_vencedor);
    else
      dbms_output.put_line('Data:
'||x.duelo_data||'; Vencedor:
'||x.tcn_vencedor);
    end if;
  end loop;
end;
```

CONSULTA USANDO SEMI-JUNÇÃO:

```
create or replace procedure Proc (produto varchar2) is
  cursor Cursor1 is
    select T.tcn
    from Treinador T
    where exists (select * from Item I
    where T.tcn = I.comprador and I.nome_do_Item =
produto);
begin
  for x in Cursor1 loop
    dbms_output.put_line('TCN: '||x.tcn);
  end loop;
end;
```

CONSULTA USANDO ANTI-JUNÇÃO:

```
create or replace function Func (mov_in varchar2, id_in number)
return varchar2 is
    result varchar2(20) := 'a';
    cursor Cursor1 is
        select P.P_id from Pokemon P
        where not exists (select A.id_Pokemon from Aprende A
        inner join Movimento M on M.codigo_Item = A.codigo_Item
        where P.P_id = A.id_Pokemon and M.nome_do_movimento
        = mov_in);
begin
    for x in Cursor1 loop
        if x.P_id = id_in then
            result := 'nao aprendeu';
        end if;
    end loop;
    if result = 'a' then
        result := 'aprendeu';
    end if;
    return result;
end;
```

CONSULTA USANDO SUBCONSULTA DO TIPO ESCALAR:

```
create or replace trigger gatilho1
after update of P_ataque on Pokemon
for each row
declare
    cursor Cursor1 is
        select P_ataque from Pokemon where
            P_ataque > (select avg(P_ataque) from Pokemon);
begin
    for x in Cursor1 loop
        if :new.P_ataque = x.P_ataque then
            dbms_output.put_line('Esse Pokemon possui um ataque
            acima da média');
        end if;
    end loop;
end;
```

CONSULTA USANDO SUBCONSULTA DO TIPO LINHA:

```
create or replace procedure Proc (tcn_in number) is
    result Cidade%rowtype;
begin
    select * into result from Cidade C
    where (C.assinatura_geologica,C.nome_da_cidade) =
        (select T.cidade_ass_geo, T.cidade_nome from Treinador
        T where tcn = tcn_in);
    dbms_output.put_line('Assinatura Geologica:
    '||result.assinatura_geologica);
    dbms_output.put_line('Nome: '||result.nome_da_cidade);
    dbms_output.put_line('Bioma: '||result.bioma);
end;
```

CONSULTA USANDO SUBCONSULTA DO TIPO TABELA:

```
create or replace function Func (ranking_in number)
return varchar2 is
    result varchar2(3) := 'a';
    cursor Cursor3 is
        select ranking from Treinador
        where tcn in (select tcn_vencedor from Duelo);
begin
    for x in Cursor3 loop
        if ranking_in = x.ranking then
            result := 'sim';
        end if;
    end loop;
    if result = 'a' then
        result := 'nao';
    end if;
return result;
end;
```

CONSULTA USANDO OPERAÇÃO DE INTERSECÇÃO:

```
create or replace procedure Proc (cod_in number) is
    flag number := 0;
    cursor Cursor4 is select * from
        (select * from Insignia) intersect
        (select * from Movimento);
begin
    for x in Cursor4 loop
        if x.codigo_da_insignia = cod_in then
            flag := 1;
            dbms_output.put_line('Parabens, voce ganhou o
            brinde!');
        end if;
    end loop;
    if flag = 0 then
        dbms_output.put_line('Voce nao ganhou o brinde :(');
    end if;
end;
```